



Otimização de um sistema flexível de fabrico usando virtualização num contexto de digital twin

Mathaeus Lukas Franklin Rutherford T M de Souza

42829

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Engenharia Industrial Eletrotécnica.

Trabalho orientado por:

Paulo Leitão

Joaquim de Mira Júnior

Bragança

2020



Otimização de um sistema flexível de fabrico usando virtualização num contexto de digital twin

Mathaeus Lukas Franklin Rutherford T M de Souza

42829

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para
obtenção do Grau de Mestre em Engenharia Industrial Eletrotécnica.

Trabalho orientado por:

Paulo Leitão

Joaquim de Mira Júnior

Bragança

2020

Agradecimentos

Agradeço ao professor Joaquim de Mira Júnior por me apoiar e me mostrar oportunidades de crescimento, tanto profissional como pessoal.

Ao professor Paulo Leitão pela orientação e compartilhamento de seu vasto conhecimento.

Ao professor José Lima pelo apoio e energia investidos em mim.

Aos colegas de CeDRI pela convivência, tempo e compreensão.

Ao IPB e UTFPR por me possibilitar seguir meus sonhos.

Agradeço especialmente à minha família e amigos por todo amor compartilhado.

Resumo

A Indústria 4.0 utiliza ferramentas em desenvolvimento para concretizar uma transformação industrial baseado nos conceitos de *Internet of Things* (IoT), gerando impactos reais sobre atividades dinâmicas e sistemas de produção flexíveis. Assim, o *Digital Twin* possibilita a digitalização dos processos industriais através de simulação em ambientes 3D que recebem os dados do sistema físico, realizando assim monitoramento e controle.

Pela necessidade de especificação dos produtos, redução da utilização de recursos, otimização dos processos e maior precisão nas tomadas de decisão, foi desenvolvido um *Digital Twin* baseado no ambiente físico flexível de fabrico composto por um conjunto de estações Fischertechnik, um robô IRB 1400 ABB e um PLC de Omron e Schneider.

O modelo virtual foi criado em plataforma de simulação de eventos discretos SIMIO, na qual as lógicas de operação foram desenvolvidas, e alimentado com as informações recebidas das células físicas através de uma aplicação de aquisição de dados em JAVA, sendo assim possível a visualização dos níveis produtivos das simulações em uma interface interativa criada através da plataforma NodeRed.

Os resultados encontrados foram satisfatórios na medida em que foi possível a tomada de decisão pelo utilizador, incluindo a possível seleção de cenários alternativos para a melhor performance do processo produtivo, considerando análise de lucratividade, tempo de operação de cada ciclo de produção flexível, utilização de recursos e sua correlação direta com os custos gerados.

Palavras-chave: Digital Twin, Indústria 4.0, Simulação, Manufatura, IoT, SIMIO.

Abstract

Industry 4.0 uses a set of tools under development to achieve an industrial transformation based on the concepts of Internet of Things (IoT), generating real impacts on dynamic activities and flexible production systems. In this way, Digital Twin enables the digitisation of industrial processes through simulation in 3D environments that receive data from the physical system, thus performing monitoring and controlling.

Due to the need for product specification, reduced usage of resources, optimization of processes and greater precision in decision making, a Digital Twin was developed based on the flexible physical manufacturing environment composed of a set of Fischertechnik stations, a IRB 1400 ABB robot and an Omron and Schneider PLC.

The virtual model was created in a SIMIO discrete event simulation platform, in which the operating logic was developed, and fed with the information received from the physical cells through a data acquisition application in JAVA, thus making it possible to view the levels simulations in an interactive interface created using the NodeRed platform.

The results obtained were satisfactory in a way that it was possible for the user to act with improved decision-making capability, including the possible selection of alternative scenarios for the best performance of the production process, considering profitability analysis, operating time of each flexible production cycle, usage of resources and their direct correlation with the costs generated.

Keywords: Digital Twin, Industry 4.0, Simulation, Manufacturing, IoT, SIMIO

Conteúdo

1	Introdução	1
1.1	Justificativa	3
1.2	Objetivos	3
1.3	Estrutura do Documento	4
2	Referencial Teórico	7
2.1	Indústria 4.0 e Revoluções Industriais	7
2.2	Digital Twin	12
2.3	Desenvolvimento do Sistema de Manufatura para o Digital Twin	16
2.4	Integração do Digital Twin	19
2.5	Sumário	22
3	Arquitetura do Digital Twin no Sistema de Manufatura Flexível	25
3.1	Estrutura do <i>Digital Twin</i>	25
3.2	Cenário Físico	28
3.3	Conectividade	33
3.4	Repositório de Dados	37
3.5	Simulação	37
3.6	Monitoramento	38
3.7	Sumário	39
4	Implementação do <i>Digital Twin</i> no Sistema de Manufatura Flexível	41

4.1	Desenvolvimento da Aquisição e Comunicação de Dados	41
4.2	Desenvolvimento da Simulação	47
4.3	Desenvolvimento do Monitoramento	53
4.4	Sumário	55
5	Resultados e Discussões	57
5.1	Aquisição e Comunicação de Dados	57
5.2	Cenários Simulados	58
5.3	Simulação	59
5.4	Monitoramento	63
5.5	Dificuldades de Desenvolvimento	64
5.6	Sumário	66
6	Conclusões e Trabalhos Futuros	67

Lista de Tabelas

3.1	Características RTU/TCP Modbus.	33
5.1	Resultados do estudo de caso: relação com utilização de recursos.	60
5.2	Resultados do caso de estudo: relação com a receita.	62

Lista de Acrônimos

AI	Artificial Intelligence
API	Application Programming Interface
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CC	Corrente Contínua
CeDRI	Centro de Pesquisa em Digitalização e Robótica Inteligente
CPS	Cyber-Physical System
DT	Digital Twin
GUI	Graphical User Interface
HMI	Human-Machine Interface
IDE	Integrated Development Environment
INWG	International Network Working Group
IoT	Internet of Things
IP	Internet Protocol
IPB	Instituto Politécnico de Bragança
LCAR	Laboratório de Controlo, Automação e Robótica
MQTT	Message Queue Telemetry Transport
PLC	Programming Logic Controller
RFID	Radio Frequency Identification
SCADA	Supervisory Control and Data Acquisition
TCP	Transmission Control Protocol

Lista de Figuras

2.1	Evolução das revoluções industriais [1].	9
2.2	Tecnologias típicas da indústria 4.0. [7]	10
2.3	Características do sistema ciberfísico. [1]	12
2.4	Número de publicações relevantes sobre <i>Digital Twin</i> . [13]	13
2.5	Abordagem de integração do <i>Digital Twin</i> . [8]	14
2.6	Pilha de camadas de internet e o modelo de referência OSI. (Adaptado de [19]).	17
2.7	Transmissão dos dados por barramento físico ou por Ethernet.	18
2.8	Integração do <i>Digital Twin</i>	20
3.1	Arquitetura geral do <i>Digital Twin</i>	26
3.2	Sentido do fluxo de dados do <i>Digital Twin</i>	27
3.3	Arquitetura do <i>Digital Twin</i> para implementação do caso de estudo.	28
3.4	Configuração física do sistema de manufatura.	29
3.5	Fluxo de operação do robô no sistema físico.	31
3.6	Relação mestre-escravo Modbus.	32
3.7	Pilha de comunicação Modbus. [22]	34
3.8	Esquema de comunicação MQTT. [26]	35
3.9	Estrutura da mensagem MQTT. [27].	36
3.10	Classes de objetos no SIMIO. [28].	38
4.1	Endereçamento dos registradores[30].	42
4.2	Inicialização dos parâmetros utilizados por Modbus.	43
4.3	Conexão TCP-Master que comunica os módulos.	43

4.4	Procedimento de parametrização MQTT.	45
4.5	Tratamento dos dados comunicados pelo tópico em MQTT.	46
4.6	Acesso da informação através de tópico.	47
4.7	Circuito lógico para a operação de duas máquinas <i>punching</i>	48
4.8	Circuito lógico que relaciona a operação de duas máquinas <i>indexing</i>	49
4.9	Circuito lógico: operação máxima de duas máquinas <i>indexing</i> e duas <i>punching</i>	49
4.10	Representação 3D dos elementos de uma configuração produtiva.	50
4.11	Processo de tomada de decisão do robô manipulador no ambiente de simulação.	51
4.12	Processos relacionados quanto à tomada de decisão do robô manipulador.	51
4.13	Tabela relacional de dados de saída.	53
4.14	Representação dos blocos programáveis em NodeRed	54
4.15	Dashboard com a configuração do cenário padrão	55
5.1	Dashboard: comparação entre a configuração básica e duas máquinas <i>punching</i>	64
5.2	Dashboard: configuração para duas máquinas <i>punching</i> e uma máquina <i>indexing</i>	65

Capítulo 1

Introdução

As etapas do desenvolvimento industrial que afetaram e ainda afetam o método produtivo são muitas vezes classificadas como "Revoluções Industriais", sendo que a mais recente é denominada Indústria 4.0 [1]. A primeira revolução industrial é, até então, considerada a mais disruptiva, uma vez que promoveu impactos extremos nas sociedades como um todo. A partir disto as mudanças nos hábitos continuaram a ocorrer e, portanto, as sociedades necessitaram de novas formas de modelos produtivos, levando a tecnologias mais modernas. Atualmente, métodos de simulação são empregados sobre os sistemas de manufatura, auxiliados pela digitalização destes ambientes através da criação do *Digital Twin* (DT), que auxilia na supervisão da produção [2][1][3].

Um aspecto de extrema relevância no contexto das evoluções é o desenvolvimento de técnicas de gestão moderna da produção, ou ainda uma iniciação à produção flexível, nomeadamente Taylorismo e, posteriormente, Toyotismo [2]. Estes métodos modernos de gestão permitiram soluções a importantes questões como alto volume de produção e variedade de produtos.

A migração das tecnologias analógicas para digitais apresentou a produção modular como substituição à produção integral, mudando drasticamente os ciclos de produção [4].

Neste contexto, além das questões de volume e variedade produtivas, foi inserido a importante questão do tempo de entrega, que exigiu imediatismo e flexibilidade produtiva. Foram desenvolvidos ainda modelos mais dinâmicos de análise e monitoramento

dos sistemas de sensoriamento, além de melhor controle dos métodos produtivos que já surgiam de forma mais descentralizada [4].

Diante do desafio da melhoria dos aspectos produtivos, inovadores conceitos e tecnologias relacionadas à chamada Indústria 4.0 estão sendo desenvolvidos a fim de permitir uma diferente e mais flexível abordagem aos métodos de manufatura. De acordo com [1], as novas tecnologias implementadas incluem:

- *Big Data* e análise,
- Robôs/sistemas autônomos,
- Realidade aumentada,
- Ciber-segurança,
- Manufatura aditiva,
- Internet das coisas industrial,
- Armazenamento em nuvem,
- Modelagem e simulação.

Baseado nisto, os conceitos de modelagem e simulação obtêm extrema importância quanto ao desenvolvimento das tecnologias modernas. Além do melhor desdobramento e adaptação dos sistemas microeletrônicos, a necessidade de descentralização dos métodos produtivos influencia no desenvolvimento de redes de informação que interconectam os subsistemas [1]. Desta forma, o conceito de *Cyber Physical Systems* (CPS) foi introduzido com a finalidade de permitir a coleta e transmissão de dados específicos para os devidos sistemas, após a aquisição por sistemas microeletrônicos de captação e controle [4].

Para possibilitar o correto funcionamento da dinâmica escolhida, é utilizado o conceito de *Digital Twin*, sendo este responsável pela representação virtual dos sistemas físicos, pelo monitoramento e supervisão dos complexos aspectos do sistema produtivo industrial

e de manufatura. A lógica de operação do DT permite que diversas opções de cenário sejam representadas, a fim de gerar formas de predição dos níveis produtivos.

Portanto, este trabalho foi concluído com o desenvolvimento de um *Digital Twin* que possui a finalidade de criar a representação digital através de um sistema de supervisão, controle e tomada de decisão sobre o sistema de manufatura flexível industrial físico.

1.1 Justificativa

A partir do surgimento do métodos de produção em massa e com uma maior gama de flexibilidade a fim de que os produtos finais sejam mais específicos e direcionados às necessidades pontuais dos consumidores, os processos fabris destinados a todas as áreas da sociedade seguem em constante evolução junto do desenvolvimento das sociedades de consumo, que adquirem novos hábitos rotineiros de acordo com a evolução das tecnologias [2].

No contexto atual há a necessidade de otimização dos processos produtivos no sentido em que os recursos utilizados devem ser minimizados visando a sustentabilidade do processo produtivo [5], sendo assim necessária a implementação de métodos mais dinâmicos e sofisticados [1], como no caso da implementação do *Digital Twin*.

Junto da maior quantidade de implementação dos processos produtivos flexíveis supervisionados, há a necessidade de tomada de decisão auxiliada por sistemas de monitoramento [1]. Desta forma, além de ser realizada a digitalização dos processos, é possibilitada a correta interpretação dos sistemas, e assim a atividade em prol da melhor eficiência do método produtivo.

1.2 Objetivos

O objetivo geral deste trabalho se encontra em implementar técnicas de otimização de processo de manufatura com a criação de um *Digital Twin* frente a um sistema flexível de fabrico composto por células Fischertechnik, presente no Laboratório de Controle,

Automação e Robótica (LCAR) do Instituto Politécnico de Bragança (IPB), realizando a aquisição de dados através de um sistema físico de sensoriamento, projetando estes dados coletados em um ambiente virtual para simulação, monitorização e supervisão.

Para a concretização deste trabalho, é necessário realizar os seguintes objetivos específicos:

- Revisão bibliográfica a fim de obter base teórica que permita a familiarização com o conceito de *Digital Twin* e que seja utilizada como base teórica da pesquisa;
- Introdução ao caso de estudo, sendo este o sistema flexível de manufatura e do modelo composto pelo *Digital Twin*;
- Elaboração de um sistema de aquisição de dados com a utilização de protocolo de comunicação e análise da informação através da utilização de *Integrated Development Environment (IDE)*;
- Representação da dinâmica do sistema de manufatura em ambiente de simulação de eventos discretos SIMIO, incluindo as características do gêmeo físico;
- Utilização da informação adquirida no SIMIO, comparando possíveis cenários de utilização para a otimização do sistema produtivo;
- Integração entre os sistemas, ajuste dos métodos e dados utilizados entre os diferentes ambientes de desenvolvimento;
- Integração com dashboard de visualização lógica que permita a correta interpretação e tomada de decisão na situação da *Human-Machine Interface (HMI)*;

1.3 Estrutura do Documento

Esta secção identifica os conceitos desenvolvidos em cada capítulo desta dissertação.

Capítulo 1: introdução, aspectos iniciais do estudo, demonstrando a importância do estudo dos sistemas flexíveis de manufatura e fabrico em geral. Conceitos relacionados a

indústria 4.0 também são introduzidos, e daí segue os conceitos de *Cyber Physical Systems*, *IoT*, *Digital Twin* e outros conceitos relacionados. Ainda neste capítulo será explicitado o problema sobre o qual o projeto irá decorrer-se e justificativa da pesquisa.

Capítulo 2: referencial teórico, apontando a contribuição bibliográfica na qual a dissertação é desenvolvida e os conceitos relacionados à proposta do projeto. Conceitos gerais sobre a indústria 4.0, a sua relevância social, uma abordagem ampla dos problemas sobre o qual o tema é desenvolvido.

Capítulo 3: a arquitetura do *Digital Twin* e o relacionamentos entre as partes integradas.

Capítulo 4: apresenta a implementação do projeto como um todo, sendo explicitadas as questões encontradas durante a modelagem dos cenários propostos e as abordagens adotadas para resolução dos problemas.

Capítulo 5: resultados e discussões sobre a aquisição de dados, os diferentes cenários simulados e a relevância das soluções adquiridas no âmbito digital quando comparadas com a situação real.

Capítulo 6: seção da dissertação que conclui e integra todo o projeto, com as conclusões. Oferece uma perspectiva que possibilita a análise da relevância do trabalho e a contribuição que esta possibilitou. Por fim, neste capítulo ainda estão presente sugestões para futuro desenvolvimento na área deste projeto.

Por fim, a bibliografia utilizada é disponibilizada.

Capítulo 2

Referencial Teórico

2.1 Indústria 4.0 e Revoluções Industriais

As revoluções industriais marcadas na história compõem um movimento disruptivo associado a mudanças bruscas na sociedade. Segundo [3], as revoluções surgiram no decorrer da história ao acompanhar uma mudança de paradigma proporcionada por alterações na forma de perceber o mundo e desenvolvimento de novas tecnologias, requerendo assim profundas alterações nos aspectos sociais e econômicos, ainda que este movimento disruptivo possa levar muitos anos para se implementar completamente.

A alteração nos hábitos humanos tem grande influência na emergência dos movimentos industriais. Ainda antes de primeira revolução industrial, a alimentação humana era baseada nos métodos de caça e coleta. De acordo com [3], a revolução agrária inciou-se com a domesticação de animais, e utilizando os esforços destes com o objetivo de otimizar a produção da época foi possível o transporte em massa de mercadorias e a maior facilidade de comunicação entre os grupos populacionais, formando assim as primeiras sociedades. Já no século XVII, segundo [6], foi iniciada a primeira revolução industrial, sendo que o método de geração de energia nesta revolução foi baseado na utilização de carvão fóssil como combustível, além da implementação do motor de vapor e a utilização de ferrovias como método de transporte de mercadorias e pessoas, o que foi aos poucos trocando a

forma de produção braçal por esforço mecânico.

A capacidade de manipulação de energia elétrica junto da aplicação deste método às tecnologias emergentes que surgiam entre 1940 e 1960 [6], como por exemplo a utilização de motores à combustão por conta da energia gerada através da manipulação do petróleo. As linhas de produção como método de organização industrial também se tornaram um parâmetro importante nesta revolução industrial [3], sendo que este método passou a permitir a real produção em massa, para os conceitos da época [1].

A terceira revolução foi iniciada a partir do advento da implementação de semicondutores, levando a desenvolvimentos na área da eletrônica e computação que permitiram a automação das linhas de produção, segundo [6], e, além disto, os conceitos de robótica e novas fontes de geração de energia foram introduzidos. Para que todos estes elementos pudessem ser integrados de forma concisa e eficiente, foi ainda desenvolvido conceitos de comunicação baseado em rede, dando início à era da internet [3].

Neste sentido, seguindo as novas necessidades apresentadas pela era atual, e o intenso ponto de virada de máxima implementação das tecnologias de informática e comunicação [1], se apresenta uma nova revolução no âmbito produtivo. O contexto da necessidade por novos métodos de produção que gerem mais eficiência no processo de fabrico para que seja possível uma maior evolução no aspecto econômico, a necessidade de menos resíduos serem gerados no contexto ambiental de forma que haja uma certa sustentabilidade quanto a utilização dos recursos finitos utilizados como matéria prima, além de impactos sociais que demandam mudança quanto a produção flexível de forma que os produtos criados contribuam com geração de valor para os consumidores e utilizadores, está em processo o que pode ser considerada a quarta revolução industrial [3]. Esta vem se formando sobre os conceitos já instaurados durante a terceira revolução [6].

Conforme [6], esta revolução foi iniciada junto com a disparada deste século e marca a chegada da era digital, como é também descrito na Figura 2.1: i) na primeira revolução, a era do vapor, que conta com sistemas de manufatura baseados na utilização de água como fonte de energia; ii) a segunda revolução inicia a produção em massa a partir da utilização de eletricidade; iii) a terceira revolução industrial introduz a era da informação,

que é marcada pela chegada de dispositivos eletrônicos e sistemas de automação; iv) a quarta e atual revolução, atuando com os conceitos de Indústria 4.0 e *Cyber Physical Systems*.

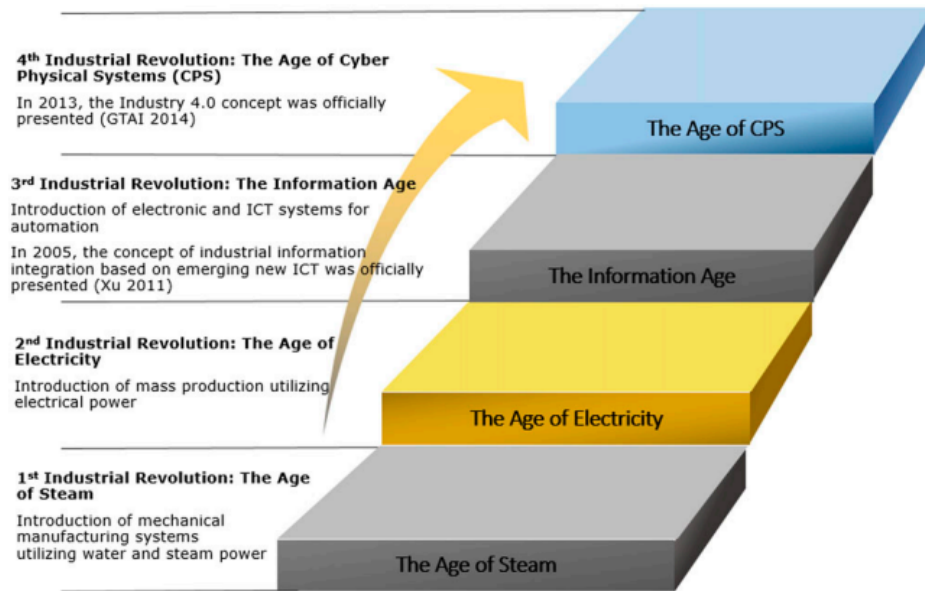


Figura 2.1: Evolução das revoluções industriais [1].

O termo Indústria 4.0 foi definido no contexto do plano de desenvolvimento tecnológico iniciado na Alemanha [6], que visa estabelecer os parâmetros sob os quais a atual revolução industrial vem a se desenvolver. Segundo [3], o termo foi estabelecido na convenção de Hannover em 2011, como uma forma de iniciar uma discussão sobre os resultados que serão gerados sobre as organizações de grande cadeia de valor através da implementação das *smart factories*. Conforme [1], a atual revolução desenvolve-se sobre IoT e *Cyber Physical Systems*, juntando os conceitos do contexto produtivo e de conectividade por rede.

Segundo [6], esta revolução tem evoluído em um ritmo exponencial em vez de linear, sendo que o principal fator de contribuição para esta situação é a necessidade quase imediata de existirem novas soluções para as questões geradas a todo momento em uma sociedade extremamente interconectada. A indústria 4.0 tem no seu escopo permitir a capacidade de integração nos níveis empresariais através da sua elevada utilização de

sistemas avançadas de comunicação, sejam estes intra ou interorganizacionais [1]. Assim, estes novos níveis de integração terão grande influência na melhoria dos indicadores de produção das indústrias de manufatura.

A aquisição da informação é decorrente dos avançados sistemas de sensoriamento em desenvolvimento desde a terceira revolução industrial e, a partir disto, um sistema interconectado é aplicado de acordo com os conceitos de IoT. Estes métodos de comunicação e integração dos módulos de uma indústria são representados de forma digital, sendo que a partir daí uma elevada capacidade de análise de dados e tomada de decisão é iniciada.

Conforme [6], junto deste movimento disruptivo estão diversas oportunidades, como a diminuição de barreiras entre os mercados, permitindo assim uma mais intensa interconectividade. Também é percebido um melhorias nas análises e tomada de decisão permitidas por conceitos de *artificial intelligence (AI)*. A partir daí, surge a implementação de diversos cenários com a criação de *designs* digitais baseados nos conceitos de *Digital Twin*.

A Figura 2.2 reúne conceitos cruciais relacionados à quarta revolução industrial.

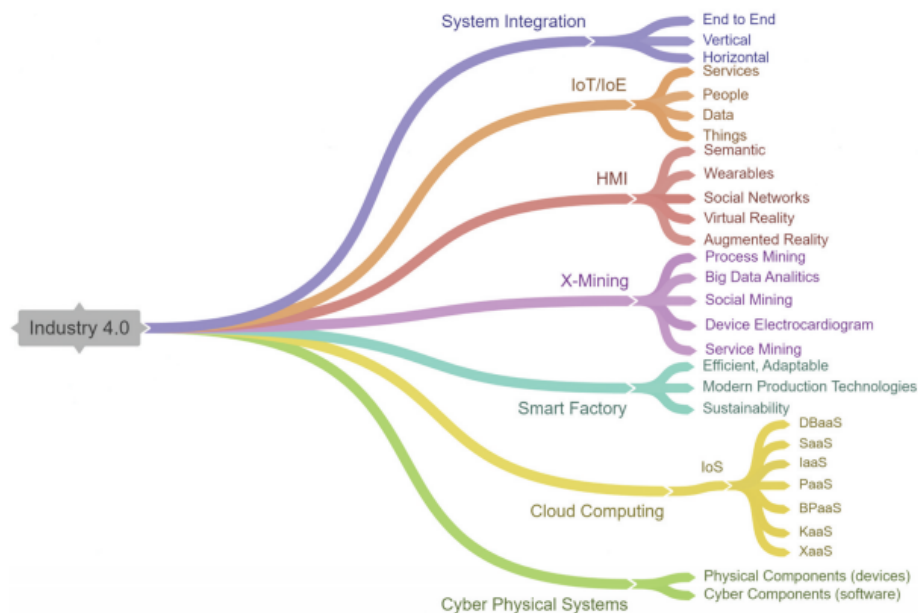


Figura 2.2: Tecnologias típicas da indústria 4.0. [7]

Como na Figura 2.2, as aplicações possibilitadas pela implementação das tecnologias

presentes no escopo da Indústria 4.0 são diversas, como listado a seguir [7].

- Manufatura aditiva que permite menores lotes de produção com características mais customizadas, reduzindo ainda custos de logística e armazenamento.
- Análise através de utilização de *big data*, ou seja, utilização de grandes quantidades de dados de diferentes fontes de informação, sustentando a possibilidade de tomada de decisão em tempo real.
- Simulação de produtos, processos, materiais através de modelos que pretendem simular as operações realizadas no âmbito físico através da criação de uma representação virtual dos elementos e processos.
- Robôs autônomos que possuem maior nível de autonomia, flexibilidade, e interação com outros robôs e humanos.
- Integração vertical e horizontal de forma que sejam criadas redes que integrem os dados de forma interna e externa.
- IoT industrial composto por dispositivos que possuem computação embarcada que podem comunicar-se em tempo real.
- Computação em nuvem que permite que a disponibilização e compartilhamento da informação entre diferentes localidades seja feita na nuvem.
- Realidade aumentada aplicada a diversas aplicações, sendo que esta tecnologia pode ser utilizada para melhorar a eficiência no ambiente produtivo, além de procedimentos de manutenção e treinamentos virtuais, por exemplo.

Como pode ser visualizado na Figura 2.3, nesta revolução industrial o setor de manufatura sofre uma quebra de paradigma com a implementação dos conceitos relacionados à aplicação do *Digital Twin*, que está incluso nos *Cyber Physical Systems*. É possível perceber a inclusão nestes sistemas de conceitos como integração de algoritmos e componentes físicos, *IoT*, comunidades inteligentes, entre outros.

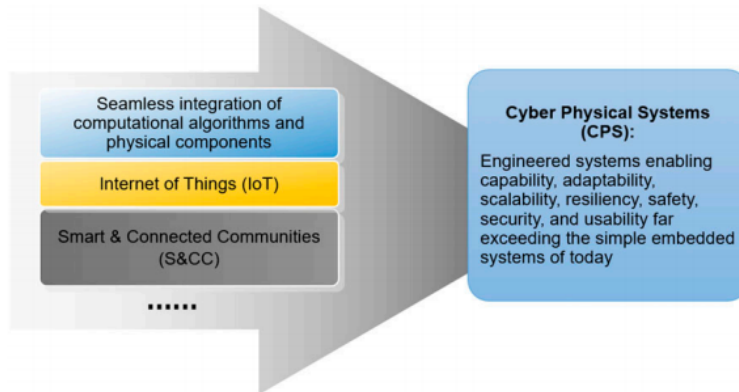


Figura 2.3: Características do sistema ciberfísico. [1]

As características cruciais que dizem respeito aos sistemas de manufatura e *Cyber Physical Systems*, e portanto devem ser levadas em conta quando ocorrer a aplicação do *Digital Twin* estão listadas na tabela 2.1.

Característica	Significado
Escalabilidade	Capacidade de alterar nível produtivo através de adição ou remoção de módulos
Conversibilidade	Possibilidade de utilizar a disposição já existente para produção de diferentes aspectos
Modularidade	Posicionamento de determinados blocos de produção em determinados esquemas operacionais
Integrabilidade	Capacidade de rápida integração de <i>hardware</i> e <i>software</i> no esquema produtivo

Tabela.2.1: Características dos sistemas de manufatura.

2.2 Digital Twin

É descrito em [8] que os conceitos relacionados às tecnologias de representação virtual datam de 2002, na época chamadas de "Mirrored Space Models". As ideias sobre espelhamento de um conjunto físico continuaram a ser desenvolvidas, sendo que em 2010 o termo

Digital Twin passou a ser associado aos modelos digitais [9].

A representação virtual dos módulos físicos, obedecendo fielmente e de forma simples os processos físicos é denominada gêmeo digital (*digital twin*), que está situada a partir do nível de comunicação, englobando ainda aspectos de nível empresarial. A definição deste paradigma envolve conceitos aplicados a simulação de dados e processos, tecnologias de processamento da informação, formas de aquisição e representação de dados [10]. A descrição desta tecnologia também foi incluída no escopo de *Cyber Physical Systems* em "Recommendations for implementing the strategic initiative Industrie 4.0", na Alemanha em 2012 [11].

Segundo [12], a definição mais comum de *Digital Twin* adotada atualmente é a proposta em 2012 como: *Digital Twin* compõe uma simulação probabilística capaz de espelhar o cenário real dos módulos físicos em múltiplas escalas e que integra diversos *designs* CAD dispostos nos ambientes. As estações compostas pelos conjuntos físicos tem o objetivo de representação dos processos de simulação com alta eficiência.

Não apenas representa as capacidades físicas do processo, mas mapeia a comunicação dos dados, possibilita previsões a partir de diferentes cenários, se adaptando às imprevisibilidades do cenário físico (*edge equipment* e *end devices*) permitindo colaboração entre os espaços físicos e digitais durante todo o ciclo de vida dos processos [10].

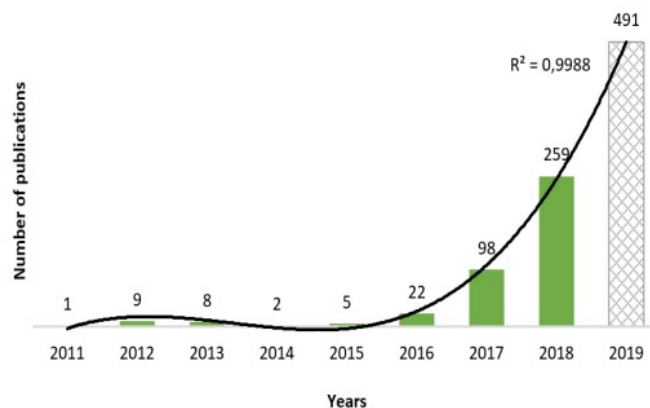


Figura 2.4: Número de publicações relevantes sobre *Digital Twin*. [13]

Com a maior implementação do conceito de *Digital Twin*, diversas publicações passaram a ser realizadas. Em [13] é realizado uma pesquisa sobre a relevância do *Digital Twin* entre 2011 e 2018, baseado no número de publicações sobre o tema. A Figura 2.4 mostra a quantidade de informação publicada sobre *Digital Twin*, demonstrando a importância adquirida por esta tecnologia durante os últimos anos.

A Figura 2.5 unifica as etapas que descrevem a elaboração de um *Digital Twin*.

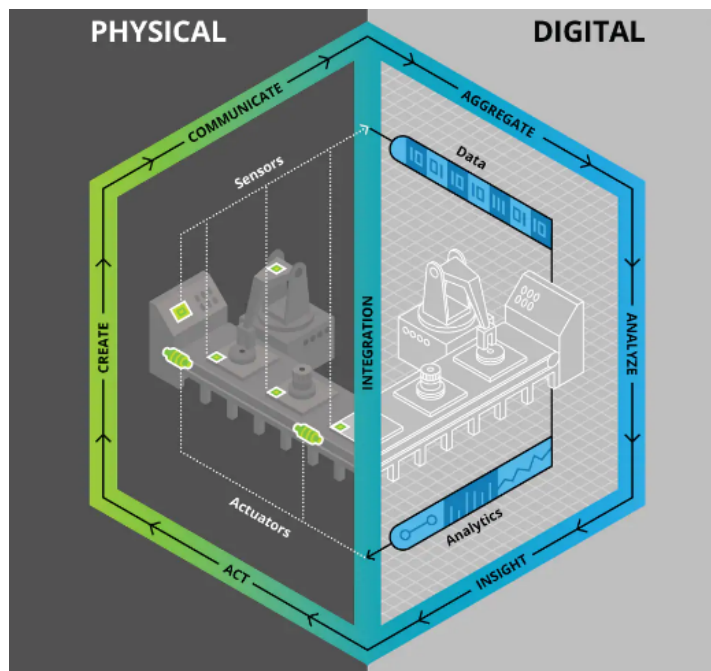


Figura 2.5: Abordagem de integração do *Digital Twin*. [8]

A partir da capacidade de atuação de sensores ao realizar a aquisição de dados, a etapa *Create* pode atuar. Em seguida, na etapa *Communication* a informação é transmitida ao *Digital Twin* através de protocolos de comunicação. É necessário ainda realizar a integração das etapas física e digital, o que é chamado de *Integration*. O passo *aggregate* realiza a disponibilização dos dados em um repositório. Em seguida, o passo *Analyze* se refere à capacidade de visualização e análise da informação processada, seguido de *Insight* que realiza a apresentação da informação. Ao contrário dos sensores, os atuadores recebem a informação e atuam. Assim, eles representam o passo *Act*.

As aplicações e benefícios proporcionados pelo *Digital Twin* englobam diversas áreas

além da manufatura, como o setor de comércio, saúde, e desenvolvimento de cidades inteligentes [13][11]. Os autores de [8] citam que por conta da capacidade do *Digital Twin* de atuar em tempo real e relacionar as diversas fases do ciclo de vida de um processo, aplicações como no setor da saúde possuem grande potencial de implementação.

De acordo com [8] o *Digital Twin* é capaz de realizar análises quanto as tomadas de decisões e otimização sobre os processos, possibilitar detalhes para a realização de manutenção preventiva, auxiliar na criação dos *designs* dos protótipos. Como exemplo, o estudo em [4] cita importantes contribuições do *Digital Twin* ao espelhar um gêmeo físico no contexto de um processo produtivo:

- Otimização do comportamento do sistema durante a sua fase de *design*.
- Compreendimento do comportamento a longo prazo e predição da atuação do processo de acordo com os fatores externos influenciadores.
- Fornecer informações continuamente no decorrer dos ciclos de vida do processo.
- Otimização dos ciclos de trabalho do sistema, baseando-se nos estados passados e presentes.
- Providenciar a confiabilidade do gêmeo físico.
- Monitoramento de deformações no material do gêmeo físico.

Ainda que as abordagens econômicas de cada ambiente industrial possuam diferentes características, em cada um desses cenários é possível identificar vantagens relacionadas a implementação de lógicas baseadas nas tecnologias possibilitadas por *Cyber Physical Systems*, como *Digital Twin*. No caso de implementação de tecnologia *Digital Twin*, de acordo com [14], tratando em termos de *Compound Annual Growth Rate – CAGR*(Taxa Compostas de Crescimento Anual), o mercado espera um crescimento de 37.87%, o que corresponde a 15.66 bilhões de dólares, até 2023. Ainda, como em [15], o mercado espera que a valorização da tecnologia passe de 3.8 bilhões de dólares em 2019, para 35.8 bilhões de dólares em 2025. Desta forma, fica claro o incremento da importância de instauração

de tecnologias baseadas em *Digital Twin* nas cadeias de valor das empresas do setor tecnológico.

Portanto, fica acertado que o conceito de *Digital Twin* inclui a capacidade de espelhamento do sistema físico [16] de forma a otimizar processos de desenvolvimento e atuação. A aplicação desta tecnologia em conceitos de manufatura passa a ter uma grande contribuição no contexto produtivo [4]. Assim, para a representação destes sistemas é necessário lançar mão de ferramentas apropriadas que representem o gêmeo físico fielmente.

2.3 Desenvolvimento do Sistema de Manufatura para o Digital Twin

Para a correta configuração do sistema físico que será representado virtualmente, é necessário lançar mão de um sistema de instrumentação baseado em sensoriamento, sistemas de medição e, se for o caso, atuadores [17][8]. O sistema de controle computacional recebe as informações destes elementos confiáveis, integrando as informações em um repositório de dados [18]. Em um primeiro nível a comunicação entre os módulos pode ser realizada a nível físico, permitindo que os dispositivos inteligentes comuniquem seus estados. No dispositivo de controle, os dados a princípio descentralizados são organizados e transmitidos em conjuntos por uma rede sem fio. É baseada em protocolos que padronizam o formato da informação, estabelecendo sistemas de hierarquia de comunicação, segurança, além de estabelecer a latência [19].

A pilha de comunicação obedece um sentido da camada de aplicação até a camada mais baixa chamada de camada física [19]. A fim de que os conjuntos de dados sejam divididos em menores grupos para que uma conexão estável e segura ocorra, além de permitir alterações nas camadas específicas sem danificar todo o conjunto através de um sistema de modularização dos seguimentos, há a distribuição em níveis de comunicação, sendo estes representados a partir das camadas situadas na Figura 2.6.

Os protocolos estabelecem os comportamentos da transmissão de dados, como no caso

de comunicações ponto-a-ponto que possibilitam que um tipo de mensagem seja retransmitido de um ponto ao outro no caso de falhas na comunicação [18]. Esta capacidade de retransmissão é utilizada na grande maioria das redes industriais aplicadas a sistemas robustos nos quais as informações não podem ser perdidas no barramento de transmissão de dados. De acordo com [19], o sistema de camadas de protocolos pode ser implementado em diferentes níveis de *software* e *hardware*.

O objetivo desta tese em aplicar a tecnologia *Digital Twin* a processos físicos requiere que alguns itens dessa pilha sejam devidamente implementados, estando estes presentes não apenas na pilha de protocolos de internet, mas também no modelo OSI de referência. Assim, o modelo OSI, descrito pela ISO em 2012 [19], é comumente utilizado por diversas aplicações baseadas na hierarquia de comunicação de internet.

Assim, como na Figura 2.6, é possível identificar que esta pilha possui dois níveis a mais, sendo estes o nível de apresentação e o de sessão, sendo que o primeiro permite a utilização de serviços que possibilitam a comunicação entre aplicações, e o segundo nível é responsável por definir a sincronização e recuperação de dados a partir de pontos chave na transmissão.

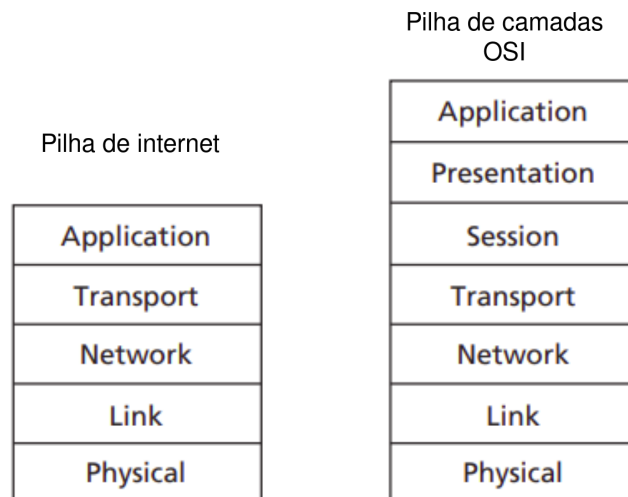


Figura 2.6: Pilha de camadas de internet e o modelo de referência OSI. (Adaptado de [19]).

Em ambas as pilhas apresentadas, a camada física tem a responsabilidade de mover

os *bits* entre os nós, ou seja, é capaz de transmitir as menores parcelas da informação entre o elemento anterior de rede e o próximo elemento [19], sendo que cada tipo de estrutura física sugere um tipo de protocolo, como por exemplo cabos coaxiais diferem seus protocolos dos cabos de fibra óptica.

No contexto ciberfísico, existem alguns nós de sensoriamento que são diretamente compatíveis com as características IP (*Internet Protocol*), enquanto outros não atendem aos requisitos e, portanto, são incapazes de comunicar os dados diretamente aos *servers* ou *brokers* utilizando apenas o protocolo de *internet*. Sendo assim, há a necessidade de transmissão dos dados aos denominados *gateways*, que possuem a habilidade de traduzir os dados recebidos pelos nós em informações compatíveis com as características IP [18], como demonstrado na Figura 2.7.

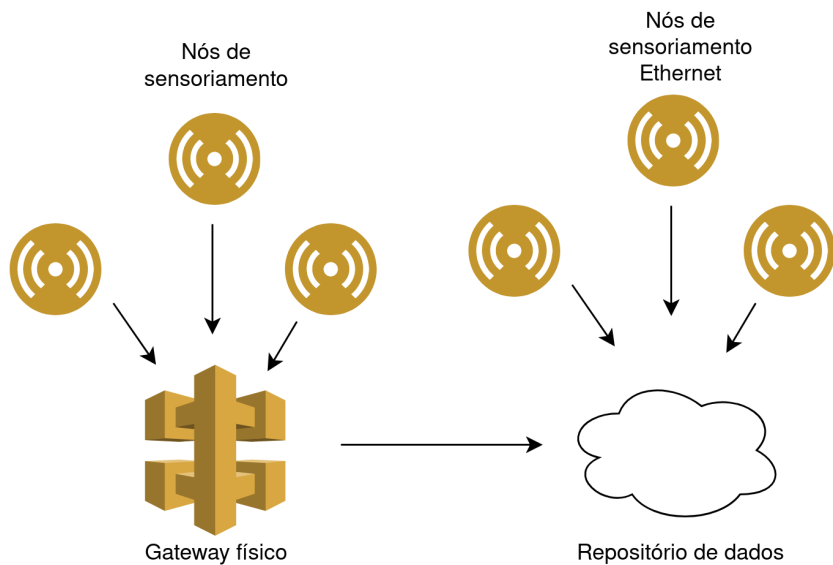


Figura 2.7: Transmissão dos dados por barramento físico ou por Ethernet.

A camada de aplicação é a responsável pela função de comunicação entre as aplicações, através de *frames*. A fim de que a comunicação entre os nós de sensores e centrais de processamento até o servidor de dados ocorra, algumas características da camada de aplicação como latência e o consumo de energia são de grande valia. Os princípios dessas arquiteturas de comunicação foram inseridos no protocolo TCP (*Transmission*

Control Protocol)[18]. Além disto, com o desenvolvimento das tecnologias, algumas das funções do protocolo TCP passaram a ser contempladas pelo protocolo IP [18], por conta da necessidade de atuação de um protocolo sem controle de fluxo e com serviços de comunicação ponto-a-ponto.

Assim, no contexto de *Cyber Physical Systems*, ao implementar um *Digital Twin* com características descentralizadas na sua configuração, os protocolos de rede são implementados a fim de comunicar os dados do gêmeo físico reunidos pelo PLC aos modelos digitais.

2.4 Integração do Digital Twin

Os dados coletados das diferentes fontes são analisados e integrados de forma a permitir a correta simulação dos modelos, o que pode ser realizado através de *softwares* auxiliares que permitem o monitoramento dos processos [10].

A partir daí, como na Figura 2.8, mecanismos como *dashboards* e tecnologias de visualização são incorporadas para que os dados coletados e tratados sejam expressos de uma forma mais explícita e que possibilite a interação humana através de interfaces homem-máquina (HMI). A utilização das interfaces gráficas simplificam a visualização das informações provindas dos sistemas, como no caso de níveis de produção, tempo de ciclo, ou ainda em relação a abordagens econômicas.

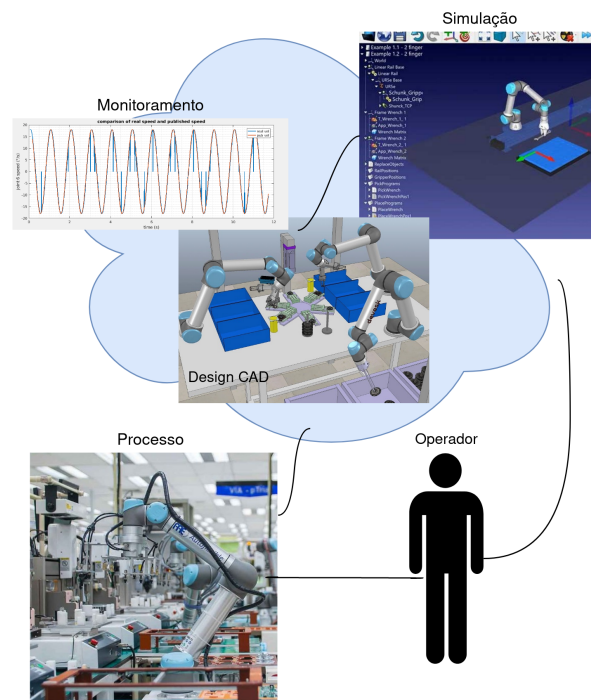


Figura 2.8: Integração do *Digital Twin*.

Com a capacidade de incluir *feedbacks* provenientes da interação com o usuário, o sistema supervisor é capaz de representar detalhes comportamentais e necessidades demonstradas pelos clientes, adaptando os processos às necessidades de mercado [11] e possibilitando atuações no setor de gestão de recursos. Portanto, a padronização das informações, como os dados de custos e lucratividade, percebidas no esquema da Figura 2.8, em um repositório de dados (*data lake*) possibilitam a correta tomada de decisão no contexto do método produtivo [8].

Os softwares simuladores auxiliam na projeção da informação reunida a procedimentos que representam o contexto produtivo, gerando resultados de monitoramento e formas de predição a partir dos cenários desenvolvidos [4].

As tecnologias que permitem a representação fiel dos módulos físicos são baseadas em modelagem 3D e sistemas CAD, que são comumente utilizadas também em outras aplicações. Os elementos virtuais representados nestes módulos incluem os modelos geométricos dos produtos, processos, usuários, ambientes e comportamentos [8]. Em relação

aos comportamentos, o módulo digital não só representa os produtos, processos e usuários individualmente, mas também a interação entre estes aspectos.

A partir da necessidade da complexa simulação e análise entre os comportamentos dispostos nos ambientes, são aplicadas técnicas de simulação [20]. Estas técnicas incluem níveis de complexidade quanto a interação dos aspectos, melhorando a capacidade de visualização, análise, validação, otimização, distribuição de diferentes tipos de cenário para se obter capacidades de previsão nos modelos criados [4][8].

Alguns dos principais *softwares* que permitem o desenvolvimento CAD e que são extensamente utilizados para inclusão nos sistemas de simulação são SolidWorks, AutoCAD, CATIA [4], sendo que para cada tipo de desenvolvimento um destes pode oferecer a melhor solução pelos seus diferentes aspectos.

Sendo assim, alguns ambientes representam de forma mais completa o sistema de produção de manufatura. São simuladores discretos, portanto permitem a supervisão dos processos no decorrer do tempo, como os ambientes estão listados a seguir:

- O *software* SIMIO (SIMulation Modeling framework based on Intelligent Objects) tem o objetivo de simulação de modelagens baseado em objetos inteligentes e pode ser utilizado para simulação, agendamento de produção, tomada de decisão em um ambiente colaborativo, além de permitir análise de riscos em tempo real e a resolução de questões mais complexas de diversos setores dinâmicos. Por conta da grande possibilidade de escalabilidade e capacidade de processamento de dados a fim de permitir abordagens matemáticas e financeiras, possibilidade de comunicação de dados com meios externos, visualização clara e em tempo real de processos, o *software* pode ser utilizado para o desenvolvimento do modelo digital baseado na configuração física, permitindo programação de seus módulos, adição de *plug-ins* ou *add-ons*.
- Visual Components *software* da empresa KUKA apresenta a capacidade de simulação de eventos voltados à manufatura. Obtendo uma gama de recursos alternativos, incluindo estações com maior especificidade, como no caso do robô KUKA, Visual

Components é capaz de ser bastante detalhista quanto ao planejamento de *layouts*, simulação dos níveis de produção, conexão com o sistema de controle e programação *off-line*.

- O *software* Arena da empresa Rockwell Automation apresenta uma opção para simulação de eventos discretos e apresenta recursos que oferecem capacidades para o desenvolvimento de automação. Através de modelos desenvolvidos pelo usuário, Arena apresenta a possibilidade de especificação no ambiente digital do processos e produtos físicos a fim de que eventos sejam simulados propriamente.
- CoppeliaSim, anteriormente V-Rep, possui um ambiente integrado de desenvolvimento baseado no desenvolvimento de arquiteturas de simulação e controle, sendo que cada componente desenvolvido pode ser continuado através de código, *plug-ins*, além de possibilidade de integração a um sistema API. Apresenta os recursos de monitoramento, supervisão e controle de sistemas.

Dessa forma, com a representação virtual do gêmeo físico através dos *designs* e a sua utilização pelos ambientes de simulação, uma importante etapa de monitoramento pode ser concretizada pelo *Digital Twin* [20].

2.5 Sumário

No sentido em que os sistemas de manufatura evoluem, surge a necessidade alinhada da melhoria na eficiência dos processos e especificação de produto. O contexto da atual revolução industrial mostra que é relevante minimizar a escassez de recursos físicos e financeiros através de sistemas que permitem a representação virtual das condições físicas, além curtos prazos de desenvolvimento e prototipagem.

Após a desenvolvimento da instrumentação para captação e integração dos dados reais, no caso de aplicações IoT alguns protocolos baseados nas pilhas de comunicação respondem em tempo real e podem se mostrar os mais recomendados para alcançar o objetivo proposto.

Assim, para o desenvolvimento do Digital Twin, após a aquisição dos dados do gêmeo físico, é importante a modelagem 3D, a representação dos processos através de simulação, tratamento da informação e disponibilização deste de forma a auxiliar na tomada de decisão dos sistemas produtivos e outras aplicações representadas pela tecnologia.

Capítulo 3

Arquitetura do Digital Twin no Sistema de Manufatura Flexível

Este capítulo apresenta a arquitetura do sistema de fabrico flexível estudado e tem grande importância para que a visualização das camadas e hierarquias relacionadas aos sistemas ciberfísicos sejam corretamente explicitadas.

3.1 Estrutura do *Digital Twin*

Este trabalho é baseado na necessidade de representação digital das configurações físicas de manufatura dispostas no LCAR, laboratório presente do CeDRI (Centro de Pesquisa em Digitalização e Robótica Inteligente), no Instituto Politécnico de Bragança. O *Digital Twin* também exige uma conectividade entre os modelos físicos e digitais, e para isto é lançado mão dos conceitos de comunicação previamente descritos para que o sistema de automação seja fielmente representado em todos os níveis. A arquitetura da Figura 3.1 representa o princípio básico de conexão das tecnologias utilizadas para a criação do sistema supervisor.

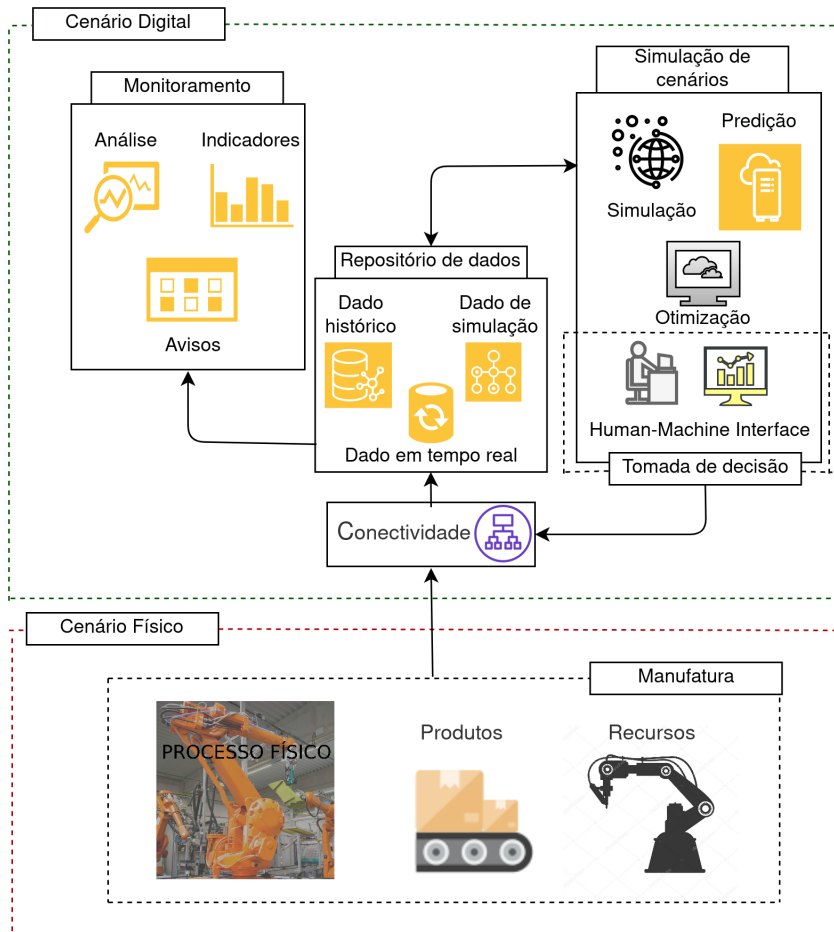


Figura 3.1: Arquitetura geral do *Digital Twin*.

Esta arquitetura divide-se em dois principais blocos, sendo eles o bloco físico e o bloco digital. Entre eles há sempre a troca contínua de informações, sendo que as interações obedecem protocolos específicos de comunicação. Ainda neste contexto, possui extrema importância o conceito de tempo de resposta dos processos e dispositivos presentes, a fim de que o contato entre as camadas da hierarquia ocorra de forma eficiente, sendo que uma falha nos tempos de execução de cada passo pode gerar uma interrupção no processo[12]. Para isso, é necessário lançar mão de ferramentas comumente utilizadas e voltadas para aplicações *IoT* e *IIoT* (*Industrial Internet of Things*) [4].

O gêmeo físico presente no laboratório é baseado em um sistema flexível de manufatura, a partir do qual pode ser criado um DT. O sistema conta com a representação

em pequena escala de um sistema produtivo dinâmico sobre o qual podem ser realizados estudos e criação de soluções de otimização. A disposição física contém "várias correias transportadoras, um robô IRB 1400 ABB, duas perfuradoras, duas linhas indexadas com 2 estações de usinagem, um centro de processamento pneumático da Fischertechnicks, vários leitores RFID, vários *buffers*. São controlados com o PLC da Omron e Schneider." [21]. Este conjunto de equipamentos permite escalabilidade de máquinas e dinamismo dos processos. As estações de operação são duas *punching* e duas *indexing*, além do robô manipulador ABB IRB 1400, e são todos controlados pelo PLC Schneider Modicon M340. A informação integrada no PLC é comunicada através do protocolo Modbus para uma aplicação de aquisição de dados em JAVA.

O conjunto de dados é disposto em uma interface de visualização no NodeRed, e utilizados para a simulação no ambiente SIMIO. O resultado gerado é apresentado em uma interface gráfica ainda no NodeRed. A partir destas informação apresentadas, o utilizador é capaz de realizar novas simulações, selecionando as características de processo na interface disponibilizada. Estes dados são comunicados ao *software* de simulação, e novos resultados são gerados. A Figura 3.2 mostra o sentido do fluxo de informação entre as aplicações, partindo do PLC.

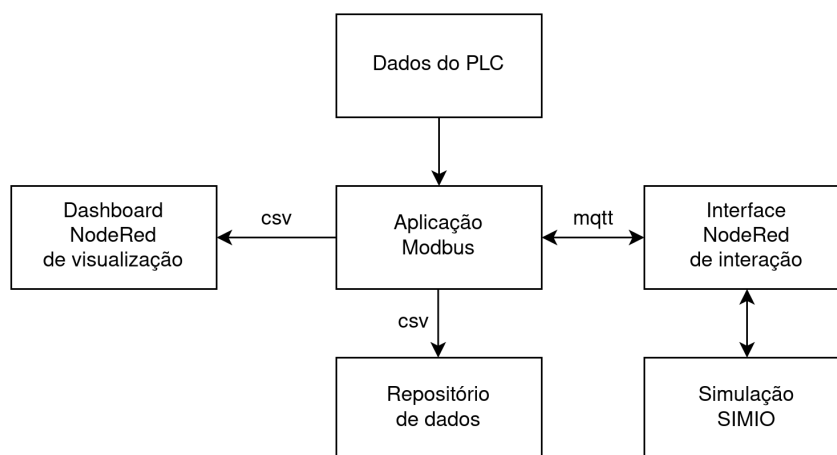


Figura 3.2: Sentido do fluxo de dados do *Digital Twin*.

Assim, a partir da visualização da informação, o utilizador é capaz de racionalizar

sobre os dados dispostos e tomar decisões sobre os processos físicos. As conexões dos módulos dispostos no *Digital Twin* com suas características específicas estão representadas na arquitetura da Figura 3.3.

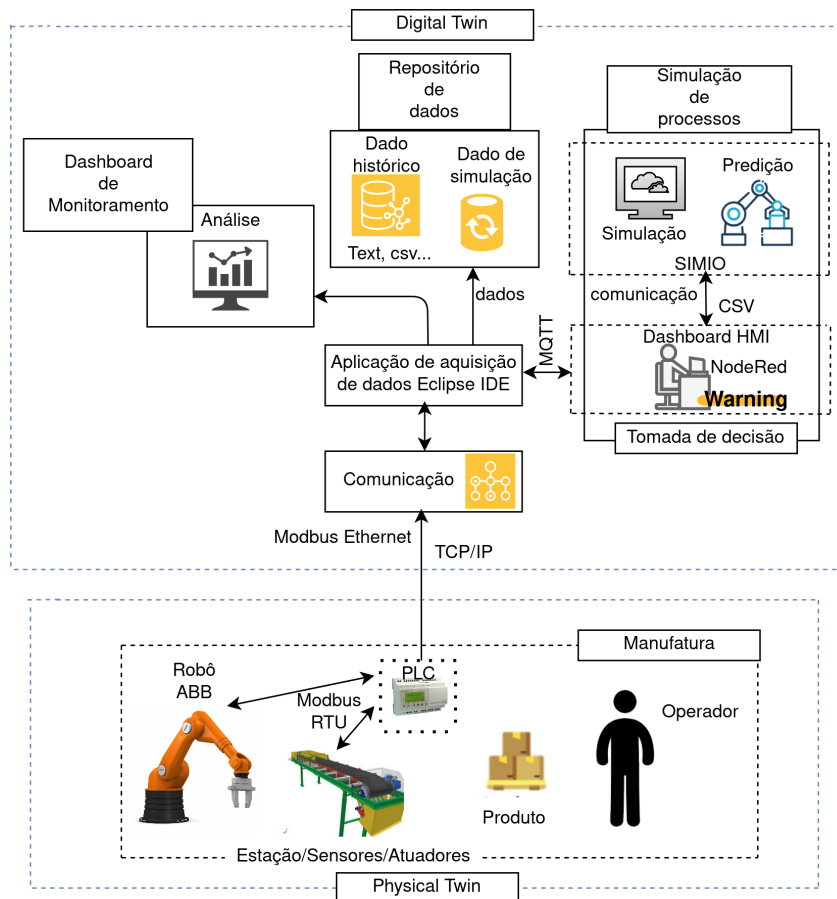


Figura 3.3: Arquitetura do *Digital Twin* para implementação do caso de estudo.

Portanto, de forma que as tecnologias selecionadas para o caso de estudo estejam explicitadas, a arquitetura da Figura 3.3, que possui as características do sistema, está devidamente descrita nas seções a seguir.

3.2 Cenário Físico

A configuração básica do *Digital Twin* é baseada na disposição de um conjunto de estações de usinagem Fischertechnik, contando com conjunto de passadeiras, além de um robô ABB

IRB 1400, junto de um sistema de controle PLC Schneider Modicon M340 programado em linguagem *Ladder*, que executa o armazenamento de todos os sinais provindos das estações em endereços de memória, utilizando então a rede interna de dados da ESTIG para comunicar.

Cada estação *indexing* possui cinco sensores fotoelétricos e oito atuadores (motor CC), além de quatro esteiras. Cada estação *punching* é equipada com dois sensores fotoelétricos e quatro atuadores (motor CC). Leitores RFID (Radio Frequency Identification) também auxiliam no sensoriamento. A estrutura física do FMS está disposta na Figura 3.4, na qual é possível perceber os itens 1 e 2 que representam máquinas *punching*, os itens 3 e 4 que contam com máquinas *indexing*, e o item 5 que é a passadeira do final de curso.

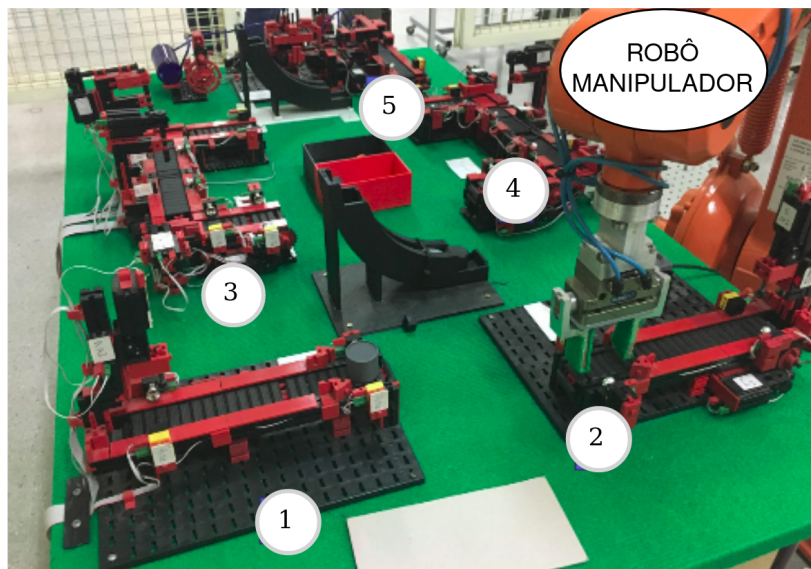


Figura 3.4: Configuração física do sistema de manufatura.

Como indicado na arquitetura 3.3, os componentes dispostos nas estações físicas como relés, sensores, atuadores, centrais de processamento e controle (PLC), ou ainda, *end devices* e *edge equipment*, compõem a estrutura física necessária de base do *Cyber Physical Systems*. Muitos destes equipamentos utilizados em aplicações *Digital Twin* estão disponíveis numa estrutura *plug and play* que permite a facilidade de integração entre os módulos. Neste sentido, os módulos físicos são representados também por elementos com

características conjuntas, como estações de manipulação do produto, elementos de transporte como passadeiras e robôs, comandados pelos PLCs que realizam o procedimento de recepção, manipulação e controle local das informações do nível.

No caso de estudo, as peças em produção são transmitidas pelas passadeiras e manipuladas entre as estações de fabrico pelo robô ABB, que é programado em linguagem RAPID. As estações também são controladas pelo PLC, e apenas diferem entre as atividades *indexing* e *punching*. A movimentação das peças pelo robô manipulador é crucial, respeitando a ordem de produção. O diagrama da Figura 3.5 representa a operação do robô no sistema, manipulando a peça entre as estações de acordo com a detecção pelo RFID e disponibilidade das máquinas disponíveis para operar.

Esta situação conta também com a presença de operadores, que realizam a visualização e programação destes módulos *plug and play* no sentido de auxiliar a tomada de decisão ainda no nível de campo. Há ainda a necessidade de gestão das ferramentas e recursos produtivos instalados como estrutura do processo através de métodos de manutenção, detecção de falhas e gerenciamento dos equipamentos.

O nível de campo (*field level*) compreende a camada mais baixa da pirâmide de automação, composto principalmente por equipamentos de sensoriamento dos sinais físico que são traduzidos em sinais elétricos, a fim de que a informação seja coletada corretamente e devidamente transmitida à central de processamento de dados através do barramento físico que permite a condução dos sinais e impulsos elétricos. O conjunto de atuadores, responsáveis pela movimentação dos dispositivos, possuem caráter hidráulico, pneumático, ou elétrico. O tempo de resposta das funcionalidades na base da pirâmide exige um grande imediatismo, sendo este estabelecido na escala de microsegundos ou milisegundos.

A saída dos sinais do nível de campo é comunicada ao nível de célula, que por sua vez compreende a capacidade de manipulação dos dados recebidos pelo nível anterior através de aplicação de controle atribuído ao PLC situado na parte física do arquitetura 3.3. Por conterem capacidade de processamento e armazenamento da informação, estes controladores realizam as tarefas previamente programadas em sua interface a fim de realizar o controle necessário sobre o fluxo de dados obtido através do sensoriamento do

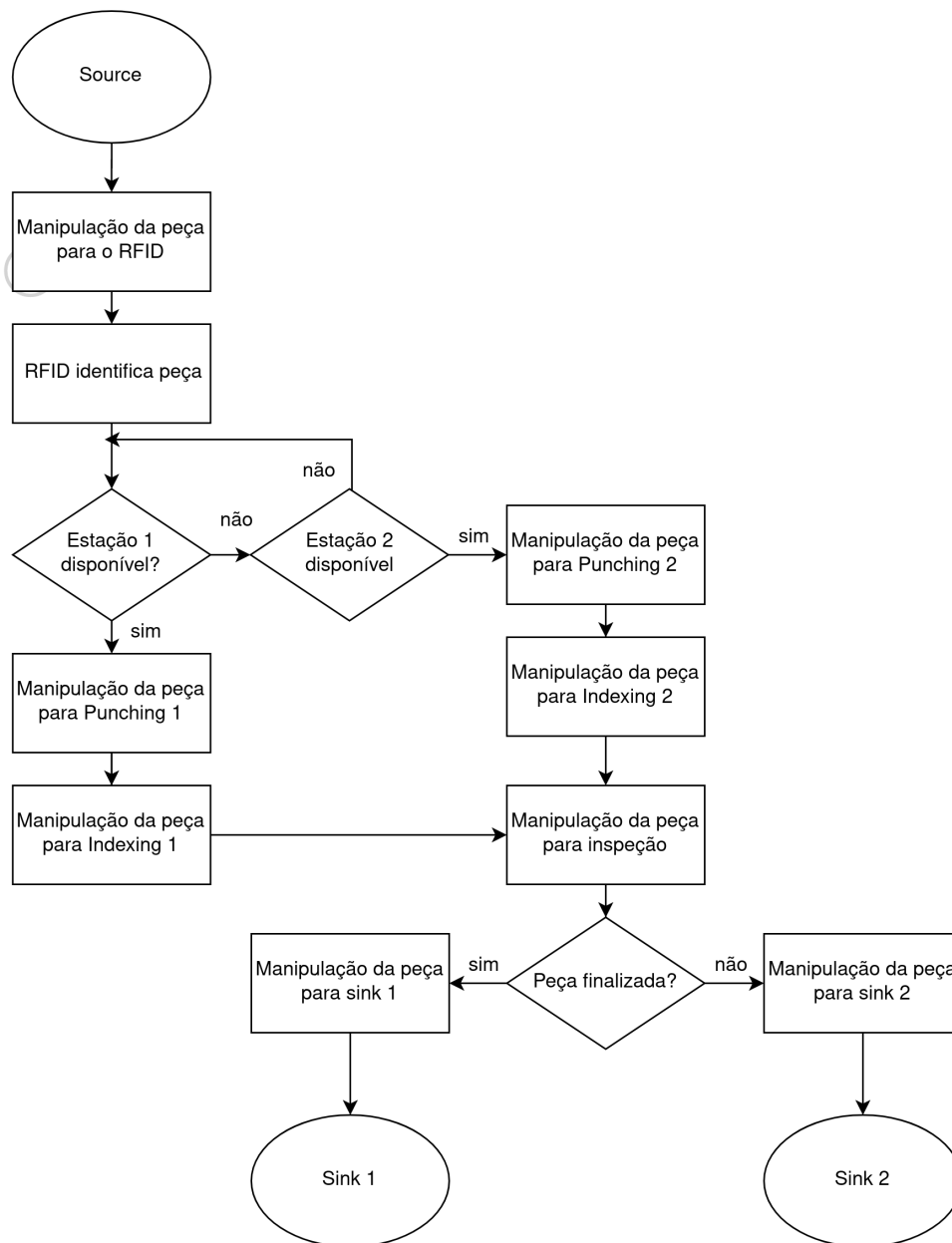


Figura 3.5: Fluxo de operação do robô no sistema físico.

nível anterior, como tempo, velocidade, temperatura, e outros tipos de alimentação de dados.

O controlador Schneider Modicon M340 utiliza de comunicação ponto-a-ponto, através de CPU e drivers RS485/422/232, sendo que uma biblioteca Modbus é necessária para padronizar a comunicação mestre-escravo entre os pontos. Este contato é realizado

em pares, sempre com um mestre presente que é responsável iniciar a comunicação. A quantidade de escravos comunicando com um mestre é definida da seguinte forma [22]:

- Apenas um escravo, utilizando RS232.
- Até 10 escravos, com RS422.
- Até 32 escravos, ao utilizar-se RS485.

Segundo [22], os nós escravos são acionados em uma determinada ordem no barramento e cada mensagem enviada recebe uma resposta, evitando que haja conflitos na comunicação. Assim, é possível associar esta informação ao *Digital Twin*, já que o protocolo disponibiliza as informações comunicadas na rede. A Figura 3.6 mostra o relacionamento mestre-escravo.

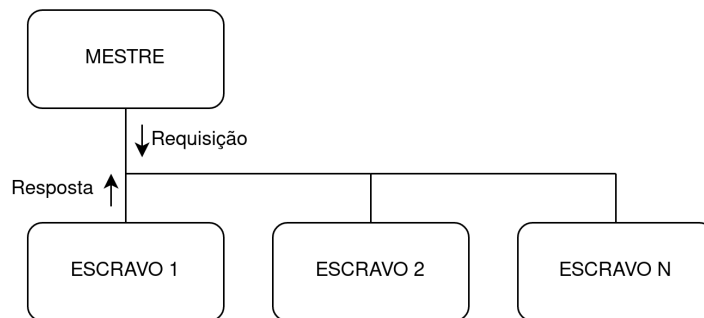


Figura 3.6: Relação mestre-escravo Modbus.

O PLC situado no LCAR é utilizado em diversas aplicações por conta da sua capacidade *plug play*, sendo que seu algoritmo executa o armazenamento dos sinais dos sensores e atuadores através de endereços de memória. No caso de estudo atual, os dados recebidos pelo PLC são transmitidos através da rede do IPB. Para a aplicação do estudo de caso em particular e para o cenário industrial é de grande valia a utilização de RS485, uma vez que este permite escalabilidade e liberdade na utilização das estações físicas.

Portanto, a primeira etapa do projeto conta com a disposição e adaptação do *layout* do sistema flexível de fabrico para que a operação do robô esteja alinhada aos procedimentos das passareiras. A informação dos estados é reunida e centralizada pelo PLC.

3.3 Conectividade

Para transmitir a informação do ambiente do gêmeo físico para o sistema virtual, é necessário estabelecer comunicações de dados através de protocolos que padronizam a forma como a informação é tratada e encapsulada.

Como uma extensão à conexão através de Modbus no nível físico, é necessário lançar mão da pilha TCP/IP do protocolo para a transmissão da informação do PLC para a rede disponível. Assim, é necessária uma aplicação de nível Ethernet ao aceder a essa informação através de uma aplicação específica em linguagem JAVA e direcionar o conjunto de dados da memória dos registradores a um conjunto de variáveis. Estes dados são disponibilizados nos bits de memórias dos registradores, e portanto a informação fica situada de forma binária, sendo acessada pelas variáveis da aplicação. A informação recolhida corresponde aos estados da linha de produção.

Modbus TCP/IP é um protocolo industrial de comunicação desenvolvido pela Schneider Electric em 1979 com a finalidade de transmitir informações de seus controladores lógicos programáveis, mas que hoje permite comunicação entre diversos equipamentos eletrônicos [23]. Pela sua facilidade de implementação, utilização de poucas restrições, além da forma de encapsular pacotes de dados, utilizando Ethernet, ou ainda o muito utilizado *driver* RS485, que estabelece os parâmetros da comunicação serial, sendo que este protocolo permite o acesso da informação por diversos dispositivos conectados simultaneamente a Ethernet. O endereçamento da comunicação Modbus é realizado respeitando a capacidade que é demonstrada na Tabela 3.1.

Estação	Endereçamento
RTU	1 até 65535
TCP	endereçamento IP/ número de Port

Tabela 3.1: Características RTU/TCP Modbus.

Por Modbus/TCP utilizar Ethernet, a velocidade de transmissão dos dados é imensamente maior do que Modbus RTU (*Remote Terminal Unit*) que utiliza os barramentos seriais e atua no nível de campo, mas com a contrapartida da pilha TCP/IP não permitir

a mesma robustez que os dispositivos de campo.

Segundo [23], aplicações remotas utilizam extensivamente este protocolo, sendo que além do tipo simples de saída física denominada *coil* que é do escopo de aplicações, e entrada denominada *discrete input* que é fornecida por um sistema de entrada/saída com 1 bit cada, Modbus também utiliza a lógica de manipulação dos registradores utilizados no *Digital Twin* desenvolvido, que necessitam de 16 bits em espaço disponível.

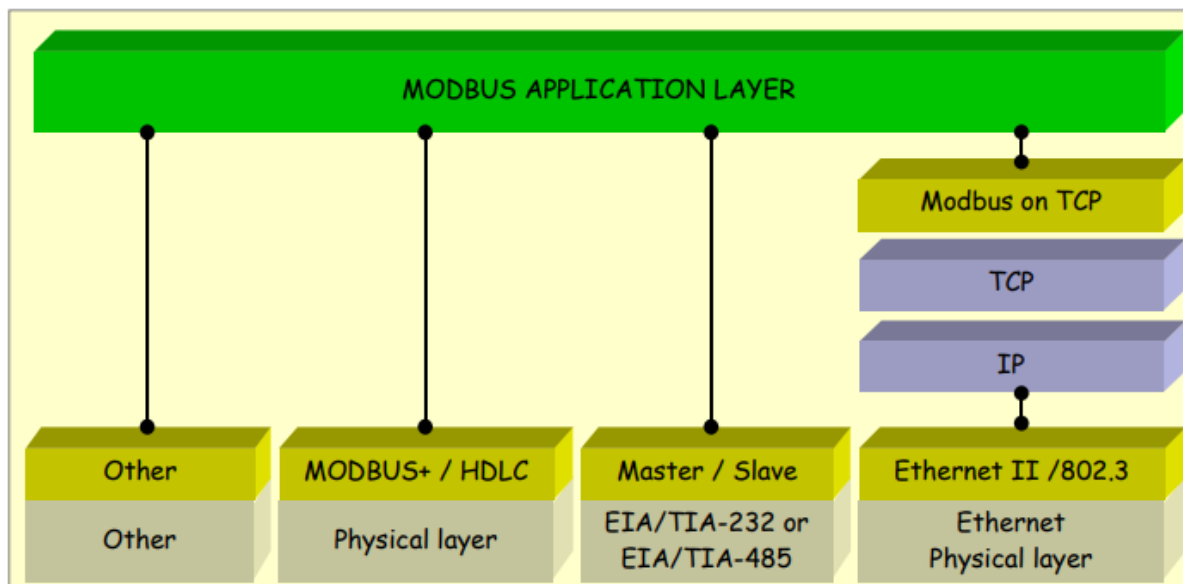


Figura 3.7: Pilha de comunicação Modbus. [22]

Como nos outros protocolos, as versões de Modbus implementam principalmente os princípios TCP e UDP[19]. No caso TCP, se conectando no *port* 502, Modbus não utiliza certas regras como *checksum* pois as outras camadas TCP já fornecem suficiente proteção. No caso da utilização de UDP, aplicações que necessitem de maior velocidade de resposta podem se beneficiar, pois neste modo o Modbus não possui cabeçalhos como em TCP[23].

Para que a informação seja transmitida da aplicação de aquisição de dados em JAVA para o ambiente de simulação do *Digital Twin*, é necessário estabelecer a forma como a informação é repassada. Uma comunicação MQTT entre a aplicação e o ambiente de monitoramento é uma possível solução no contexto do DT.

MQTT (*Message Queuing Telemetry Transport*), desenvolvido pela IBM, é um protocolo baseado no modo *publish/subscribe* como forma de troca de mensagens. Ao contrário de outras opções no mercado, é de fácil implementação, e pode ser aplicado a dispositivos "limitados" (*constrained devices*), com capacidades abrangentes em diversos cenários de atuação, como em sistemas de monitoramento e supervisão[24].

Uma vez que na utilização de sistemas *request/response* presente em outros protocolos uma grande largura de banda deve ser utilizada, além de alta necessidade de capacidades computacionais, a utilização das características *publisher/subscriber* presente em MQTT torna-se uma melhor opção para utilização em IoT, além de conter um sistema *broker* que é ponto de intermediação entre os pontos que requerem comunicação [25]. Um nó qualquer publica mensagens utilizando de tópicos específicos para o *broker*, que disponibiliza a informação. Finalmente, um servidor (*server*) inscreve-se neste tópico e o *broker* envia a informação para aquele domínio.

Uma simples esquema de funcionamento é representado na Figura 3.8, na qual o sentido de publicação e inscrição não podem ser percebidos, sendo que o *broker* é uma peça chave pois é posicionado de forma que o todo o sistema depende deste. Utilizando TCP, o MQTT também é capaz de utilizar outras características de protocolos de rede, que forneçam configurações diferenciadas, como comunicação bidirecional, redução de perdas na comunicação e uma transmissão bem ordenada [24].

Neste sentido, os níveis de QoS são aplicados para lidar com a incapacidade do TCP de impedir perdas de pacotes de dados no processo de comunicação. As opções de níveis implementadas em MQTT para a qualidade de serviço são listadas a seguir[23].

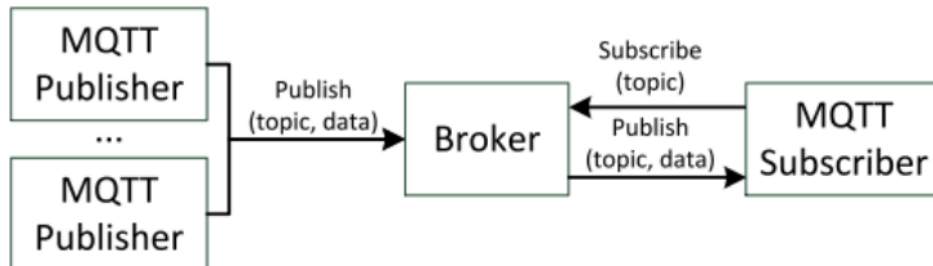


Figura 3.8: Esquema de comunicação MQTT. [26]

- QoS 0: é um nível também conhecido como "no máximo uma vez", que indica que a tentativa de entrega da mensagem é realizada com as melhores capacidades disponíveis, mas apenas uma vez. Caso o erro na entrega ocorra, a mensagem não será reenviada, o a entidade recebedora não identifica a entrega.
- QoS 1: neste nível de qualidade do serviço, a mensagem é enviada "ao menos uma vez", ou seja, a entrega é garantida. Um contraponto é que duplicatas podem ocorrer.
- QoS 2: é o nível mais eficiente de comunicação, no qual a mensagem tem garantia de ser entregue, e que a partir disto não há chances de duplicata.

bit	7	6	5	4	3	2	1	0	
byte 1	Message Type				DUP Flag		QoS Level		Ret
byte 2	Remaining length								
Variable Header									
Payload									

Figura 3.9: Estrutura da mensagem MQTT. [27].

Quanto a estrutura da mensagem, MQTT utiliza um cabeçalho fixo, seguido de um cabeçalho variável, e por fim o *payload*, como visto na Figura 3.9, sendo que a estrutura é iniciada com o cabeçalho de apenas dois bytes que, junto de uma pequena necessidade de processamento, baixa latência e pequena utilização de memória, faz com que o protocolo seja extensivamente utilizado [24].

Por fim, os dados devem ser comunicados da seguinte forma: i) os dados provindos das interfaces de interação com o usuário devem ser armazenados em arquivos CSV para serem acessados pelo software de simulação. As informações de saída da simulação de cenários são disponibilizadas neste mesmo formato de arquivo para que sejam acessados pela interface gráfica do utilizador (GUI) para realização de monitoramento por parte do supervisor; ii) na interface, por se tratar de um ambiente interativo, as entradas a serem selecionadas pelo utilizador são comunicadas através de comunicação MQTT para a aplicação em JAVA.

3.4 Repositório de Dados

O armazenamento dos dados, além da sua preservação e possibilidade de acesso são questões importantes para o ciclo de vida do caso de estudo [9]. A informação direcionada dos estados às variáveis, os dados de entrada na simulação, além dos resultados obtidos e acessados pelo sistema de monitoramento precisam estar devidamente localizados em ambiente propício ao seu acesso, com permissões e segurança, de forma que essa troca de informação ocorra sem interrupções nos tempos de ciclo do processo.

Este repositório (*data lake*) pode estar localizado em nuvem (*cloud*) ou em sistemas físicos de armazenamento [17], e possui um aspecto centralizado.

3.5 Simulação

Para que o sistema da configuração física seja representado fielmente é necessário lançar mão de uma ferramenta com base em digitalização, simulação e análise de resultados do *layout* disposto. Desta forma, a utilização de módulos digitais CAD em uma representação do sistema produtivo através de simulação mostra-se uma questão importante para a análise de processos do sistema de manufatura.

Sendo assim, o software SIMIO permite a utilização de eventos discretos que são baseados na escala temporal, possibilitando a representação da dinâmica produtiva. As ferramentas disponibilizadas pelo *software* que permitem a interação entre os módulos são denominadas objetos [28]. Estes objetos são customizáveis e são capazes de ganhar inteligência sobre a dinâmica do processo a partir da modelagem das suas características e a sua interação com o modelo desenvolvido.

As interações entre os módulos são desenvolvidas a partir da criação de um modelo com base nas classes de objetos da Figura 3.10. Além da capacidade de configuração destas características, apesar do *software* não exigir a codificação dos procedimentos através de linguagem de programação, ainda são utilizadas expressões lógicas para o relacionamento dos processos, tratamento de dados e determinados cálculos, como em relação às

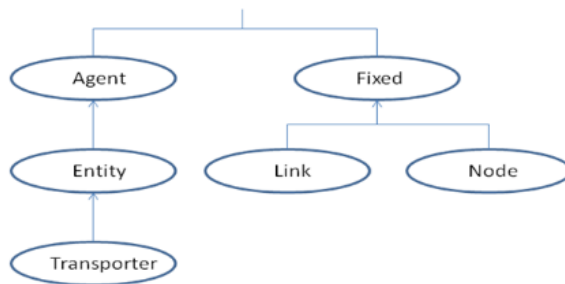


Figura 3.10: Classes de objetos no SIMIO. [28].

características de custos operacionais e tempos de ciclo.

Assim, SIMIO permite a criação de diversos cenários que representam as possíveis necessidades de produção do utilizador. Ao seleccionar as devidas características na HMI, o software deve receber os dados e realizar a simulação, por fim disponibilizando os resultados de saída do sistema digital.

3.6 Monitoramento

As características de monitoramento e supervisão são importantes para que os dados sejam visualizados em interfaces gráficas e entendidos corretamente. A ferramenta Node-RED utiliza de blocos programáveis em JavaScript em fluxos dinâmicos, conectando dispositivos de hardware, APIs e serviços online [29], a partir da estrutura Node.JS que é um *runtime* da linguagem de programação.

Nesta ferramenta podem ser desenvolvidas *dashboards* que disponibilizam a informação já tratada e permitem a interação HMI. Essa informação está disposta em formatos capazes de entendimento por parte dos operadores, baseada nos resultados dos diferentes cenários de simulação, contendo os estados dos processos físicos e possíveis previsões de funcionamento. Ao interagir com a aplicação, o usuário é capaz de interpretar as situações apresentadas, auxiliando na tomada de decisão sobre o sistema de manufatura.

Node-RED também permite que blocos que realizam a comunicação com outros softwares sejam utilizados [29]. É de grande relevância o bloco que realiza a comunicação MQTT

com a aplicação JAVA, possuindo um sentido bidirecional. Os dados recebidos da interação HMI também devem ser transmitidos ao software de simulação, e posteriormente os resultados devem ser importados em forma de arquivos de texto (CSV).

Desta forma, ao tomar as decisões necessárias, o operador é capaz de simular os cenários. A partir disto a informação é transmitida à aplicação JAVA através do bloco de comunicação MQTT.

3.7 Sumário

A arquitetura apresentada neste capítulo representa os blocos necessários para a implementação do DT. O cenário físico é conectado ao digital através de meios de comunicação que obedecem o protocolo Modbus. A informação comunicada é transmitida a um repositório de dados para que as aplicações a acessem, importando os arquivos ou através de comunicação MQTT. Aplicações são desenvolvidas com a informação adquirida, realizando simulações de cenários para predição e tomada de decisão, além de monitoramento dos estados produtivos.

Capítulo 4

Implementação do *Digital Twin* no Sistema de Manufatura Flexível

Neste capítulo estão descritas as etapas específicas do caso de estudo para a implementação do *Digital Twin*, contendo os parâmetros do desenvolvimento da transmissão de dados, do ambiente de simulação e do ambiente de monitoramento.

4.1 Desenvolvimento da Aquisição e Comunicação de Dados

Para que os dados fossem recebidos do sistema de sensores e atuadores foi utilizada comunicação Modbus/TCP através da rede interna do IPB. A aquisição destes sinais e distribuição da informação nos registradores do PLC foi baseado em um trabalho desenvolvido em outra dissertação de mestrado [30] realizada sobre o mesmo sistema físico do LCAR. A Figura 4.1 mostra a forma como o endereçamento de memória foi realizado.

Para a aquisição dos dados e possibilidade de manipulação dos *Holding Registers* de acordo com os parâmetros do protocolo, foi lançado mão da aplicação desenvolvida em JAVA através do software Eclipse, como na Figura 4.2. Uma vez que a conexão é concluída, a aplicação utiliza de uma função que distribui os registradores em uma lista,

%mw	Descrição	Célula	%mw	Descrição	Célula
Atuadores			Sensores		
0.0	Motor Tapete Tras	PB	20.0	Sensor 1 Tapete	PA
0.1	Motor Tapete Frente	PB	20.1	Sensor 2 Tapete	PA
0.2	Motor Broca Down	PB	20.2	Sensor Broca Down	PA
0.3	Motor Broca Up	PB	20.3	Sensor Broca Up	PA
0.4	Motor Tapete Frente	PA	20.4	Sensor 1 Tapete	PB
0.5	Motor Tapete Tras	PA	20.5	Sensor 2 Tapete	PB
0.6	Motor Broca Down	PA	20.6	Sensor Broca Down	PB
0.7	Motor Broca Up	PA	20.7	Sensor Broca Up	PB
0.8	Motor Tapete 1	IA	20.8	Sensor 1 Tapete 1	IA
0.9	Motor Tapete 2	IA	20.9	Sensor 2 Tapete 1	IA
0.10	Motor Tapete 3	IA	20.10	Sensor Broca 1	IA
0.11	Motor Tapete 4	IA	20.11	Sensor Broca 2	IA
0.12	Motor Empura 1 Frente	IA	20.12	Sensor Tapete 4	IA
0.13	Motor Empura 1 Tras	IA	20.13	Sensor Empurra 1 Frente	IA
0.14	Motor Empura 2 Frente	IA	20.14	Sensor Empurra 1 Tras	IA
0.15	Motor Empura 2 Tras	IA	20.15	Sensor Empurra 2 Frente	IA
1	Motor Broca 1	IA	21.0	Sensor Empurra 2 Tras	IA
1.1	Motor Broca 2	IA	21.1	Sensor 1 Tapete 1	IB
1.2	Motor Tapete 1	IB	21.2	Sensor Empurra 1 Frente	IB
1.3	Motor Tapete 2	IB	21.3	Sensor Tapete 4	IB
1.4	Motor Tapete 3	IB	21.4	Sensor Empurra 2 Frente	IB
1.5	Motor Tapete 4	IB	21.5	Sensor Broca 1	IB
1.6	Motor Empura 1 Frente	IB	21.6	Sensor Empurra 1 Tras	IB
1.7	Motor Empura 1 Tras	IB	21.7	Sensor Empurra 1 Tras	IB
1.8	Motor Empura 2 Frente	IB	21.8	Sensor Broca 2	IB
1.9	Motor Empura 2 Tras	IB	21.9	Sensor 2 Tapete 1	IB
1.10	Motor Broca 1	IB			
1.11	Motor Broca 2	IB			

Figura 4.1: Endereçamento dos registradores[30].

armazenando nas variáveis a informação de estado recebida através da comunicação. A aplicação importa algumas bibliotecas já desenvolvidas para comunicação e tratamento de erros.

Como demonstrado na Figura 4.2, a princípio é necessário estabelecer a comunicação da aplicação em JAVA com o PLC para que a informação seja transmitida, sendo que Modbus é um protocolo dinâmico e baseado em *coils*. Assim, o código é iniciado com a definição do ponto de partida para a leitura dos endereços, seguido da quantidade de registradores que foram utilizados.

Como método padrão, são instanciadas conexões TCP-Mestre, TCP-Transação, e *request/response* para os registradores. Além disto são definidas as variáveis para armazenamento dos endereços em lista e definição de métodos auxiliares.

No *script* é definido o endereço IP para a conexão Modbus do PLC e a partir disto é realizada a conexão *TCP-Master* descrita no capítulo anterior, como na Figura 4.3. O

```

1 package DT_App;
2
3 import org.eclipse.paho.client.mqttv3.MqttException;
38
39
40 public class DataProcessment {
41
42     static int START_READING_ADDRESS = 0; //used when initializing the connection
43     static int NUM_OF_REGISTERS_TO_READ = 22;
44
45
46     static TCPMasterConnection con = null;
47     static ModbusTCPTransaction trans = null;
48     static ReadInputRegistersRequest req = null;
49     static ReadInputRegistersResponse res = null;
50
51
52     /* Variables for storing the parameters */
53
54     static InetAddress addr = null; // the slave's address
55 // static int port = Modbus.DEFAULT_PORT;
56     static int ref = 0; // the reference; offset where to start reading from
57     static int count = 0; // the number of DI's to read
58     static int repeat = 1; // a loop for repeating the transaction
59 // static List<Integer> enderecos = new ArrayList<>();
60     static List<Integer> addressAcq = new ArrayList<>();
61
62     /*

```

Figura 4.2: Inicialização dos parâmetros utilizados por Modbus.

bloco A dispõe dos parâmetros de IP e a função para conexão (`con.connect()`); o bloco B possui a requisição realizada para a aquisição dos endereços de memória.

```

95     public static void main(String[] args) throws InterruptedException {
96
97         //CODE below will be commented for using the registers instead
98
99
100
101         try {
102             addr = InetAddress.getByName("192.168.38.125");
103         } catch (UnknownHostException e1) {
104             // TODO Auto-generated catch block
105             e1.printStackTrace();
106         }
107         con = new TCPMasterConnection(addr);
108
109         try {
110             con.connect();
111         } catch (Exception e) {
112             // TODO Auto-generated catch block
113             e.printStackTrace();
114         }
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 4.3: Conexão TCP-Master que comunica os módulos.

Portanto, após a conexão dos dados através de variáveis na aplicação JAVA as outras aplicações desenvolvidas podem interagir a proceder com a troca de informações. Os dados são exportados da IDE (*Integrated Development Environment*) para o repositório de dados local. O software NodeRED importa os dados do cenário padrão e os mostra na sua *dashboard* para o utilizador. Após o operador selecionar as novas necessidades produtivas e o cenário adequado, a informação é exportada para o software SIMIO, que realiza as simulações. Os dados de saída são exportados de volta para a interface gráfica com os novos resultados.

Uma aplicação foi desenvolvida estabelecendo a comunicação MQTT e transmite os dados entre o *software* Eclipse e a plataforma NodeRED de uma forma bidirecional, fazendo com que as opções de produção simuladas e selecionadas pelo usuário sejam transmitidas à aplicação JAVA. Como evidenciado no Capítulo 3, a comunicação MQTT utiliza o *broker* para setar um tópico. Por a comunicação ser bidirecional, a aplicação na IDE Eclipse pode inscrever-se no tópico e captar a informação publicada pelo bloco MQTT no NodeRed. A implementação do código para a comunicação MQTT entre as interfaces foi baseada em algumas classes e a partir disto utilizada a característica de recursividade do método de programação.

No item A do código da Figura 4.4 é definido o tópico através do qual serão utilizados *publish* e *subscribe* para a transmissão da informação. No item B foi definido o parâmetro para qualidade do serviço (QoS) como zero, pois a mensagem é enviada no máximo uma vez a fim de evitar repetições. É definido o endereço IP da CPU conectada na rede junto da extensão 1883, que define o servidor a ser utilizado. A Figura 4.4 mostra que após a realização deste procedimento, no item C é estabelecido o *CallBack*, o que garante que a conexão tenha sido realizada, e a confirmação se a conexão está operando.

```

1 package DT_App;
2
3 import java.io.UnsupportedEncodingException;
14
15 public class Application implements MqttCallback, IMqttActionListener {
16     public static final String COMMAND_KEY = "COMMAND";
17     public static final String COMMAND_SEPARATOR = ":";
18     public static final String
19         GET_DATA_COMMAND_KEY = "GET_DATA";
20     // topic name
21     public static final String TOPIC =
22         "eclipse/App";
23
24     public static final String ENCODING = "UTF-8";
25
26     // Quality of Service = at most once
27     // I want to receive all messages once
28     public static final int QUALITY_OF_SERVICE = 0;
29     protected String name;
30     protected String clientId;
31     protected MqttAsyncClient client;
32     protected MemoryPersistence memoryPersistence;
33     protected IMqttToken connectToken;
34     protected IMqttToken subscribeToken;
35
36     public Application (String name) { this.name = name; }
37
38     public String getName() { return name; }
39
53
54     memoryPersistence =
55         new MemoryPersistence();
56     String serverURI = "tcp://192.168.217.210:1883";
57     // "tcp://iot.eclipse.org:1883";
58     clientId = MqttAsyncClient.generateClientId();
59     client = new MqttAsyncClient(
60         serverURI, clientId,
61         memoryPersistence);
62     // I want to use this instance as the callback
63     client.setCallback(this);
64     connectToken = client.connect(
65         options, null, this);
66     } catch (MqttException e) {
67         e.printStackTrace();
68     }
69     }
70
71     public boolean isConnected() {
72         return (client != null) &&
73             (client.isConnected());
74     }
75
76     @Override
77     public void connectionLost(Throwable cause) {
78         // The MQTT client lost the connection
79         cause.printStackTrace();
80     }
81

```

Figura 4.4: Procedimento de parametrização MQTT.

O tratamento da conexão é realizado ao utilizar-se alguns métodos para parametrização, e após este procedimento a informação recebida no tópico é tratada por um matriz de valores, que converte dados e separa as informações, como na Figura 4.5.

```

public void messageArrived(String topic,
                           MqttMessage message) throws Exception
{
    // A message has arrived from the MQTT broker
    // The MQTT broker doesn't send back
    // an acknowledgment to the server until
    // this method returns cleanly
    if (!topic.equals(TOPIC)) {
        return;
    }

    String messageText =
        new String(message.getPayload(), ENCODING);
    System.out.println( String.format(
        "%s received %s: %s", name, topic,
        messageText));
    String[] keyValue =
        messageText.split(COMMAND_SEPARATOR);
    if (keyValue.length != 3) {
        return;
    }
    if (keyValue[0].equals(COMMAND_KEY) &&
        keyValue[1].equals(
            GET_DATA_COMMAND_KEY) && |
        keyValue[2].equals(name))
    {
        // Process the get data command

        System.out.println(String.format("%s VALUES: %f", name, message));
    }
}

```

Figura 4.5: Tratamento dos dados comunicados pelo tópicos em MQTT.

Neste *Digital Twin* em particular, como na Figura 4.6, a informação que foi enviada ao tópicos pelo *publisher* é acessada pelo *subscriber* de forma que possa ser atribuída às variáveis de processo.

No caso de estudo, o tratamento da informação através de lógicas de programação e expressões é realizado no software de simulação SIMIO, não necessitando deste tipo de tratamento de dados nas aplicações em JAVA.

```
683 Application application1 = new Application("[Info NodeRed #1]");
684 application1.connect();
685 Application masterApplication = new Application("[Info Master NodeRed]");
686 masterApplication.connect();
687
688 try {
689     while (true) {
690         try {
691             Thread.sleep(5000);
692
693             if (application1.isConnected()) {
694
695                 /* code in here will attribute info the physical module (modbus communication)
696                 */
697                 masterApplication.publishCommand(
698                     Application.GET_DATA_COMMAND_KEY, application1.getName());
699
700
701                 System.out.println("MODBUS operating...");
702
703
704                 System.out.println("Data sent to the physical modules");
705
706             }
707
708             }catch(InterruptedExcepcion e) {
709                 e.printStackTrace();
710             }
711         }

```

Console

```
DataProcessment [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (09/09/2020 11:59:02)
[Info Master NodeRed] received eclipse/App: 30;1;1;1;0
[Info Master NodeRed] received eclipse/App: 60;1;1;0;1
[Info Master NodeRed] received eclipse/App: 60;1;1;0;1
[Info Master NodeRed] received eclipse/App: 60;1;1;0;1
```

Figura 4.6: Acesso da informação através de tópicos.

Portanto, após o estado da informação junto da seleção de cenários ser importado pelo simulador, o software pode prosseguir os procedimentos de simulação.

4.2 Desenvolvimento da Simulação

A configuração do sistema de manufatura disposto possibilitou a sua representação digital através de um sistema de modelagem e simulação. Para gerar formas de predição dos níveis produtivos, a lógica de operação do *Digital Twin* permitiu que diversas opções de cenário fossem representadas. A lógica de utilização dos módulos baseia-se na característica de implementação de escalabilidade, ou seja, a possibilidade de alteração, adição, ou remoção de módulos de processamento. Sendo assim, algumas possibilidades de configuração produtiva foram representadas.

No caso de apenas uma máquina estar a produzir, a ordem é iniciada e a peça é transferida para a devida máquina em processamento, e deste ponto é transferida para o final de curso. Ao ser utilizada uma dinâmica com mais módulos em processamento,

a lógica do sistema realiza as devidas verificações, como na Figura 4.7, que representa a ordem lógica de operação quanto ao funcionamento de até duas máquinas *punching* na configuração do sistema.

Nesta situação os blocos condicionais referem-se à possibilidade das máquinas disponíveis operarem: primeiro, verifica se há alguma máquina *punching* disponível; depois, verifica qual delas está disponível e direciona a peça para a devida máquina. Deste ponto a peça é levada ao final de curso.

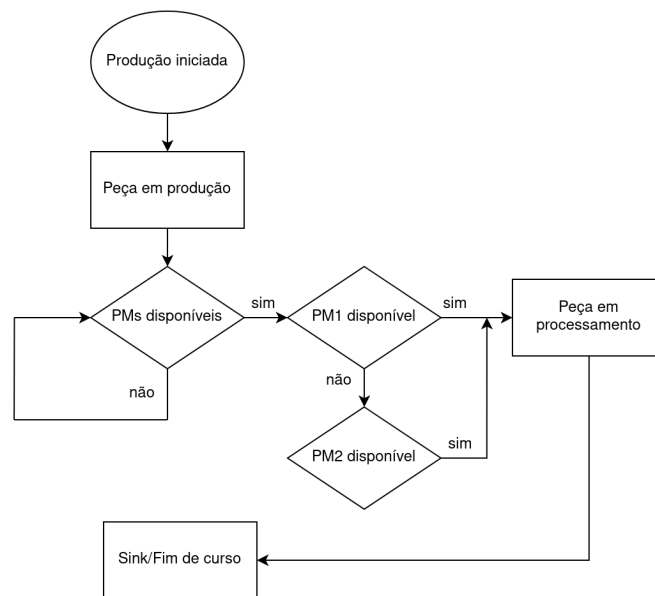


Figura 4.7: Circuito lógico para a operação de duas máquinas *punching*.

Já o fluxograma da imagem 4.8 representa de forma semelhante a ordem lógica para o funcionamento de até duas máquinas *indexing* no cenário de produção: verifica se há máquinas disponíveis, e logo em seguida direciona a peça para a máquina que pode processar naquele momento. Assim, é finalizada a operação no *Sink*.

O cenário de total operação das máquinas do sistema produtivo de manufatura conta com duas máquinas *indexing* e duas máquinas *punching*.

O fluxograma da figura 4.9 representa a sua ordem lógica de funcionamento: primeiramente a atividade *punching* é requisitada, portanto há a verificação de qual das máquinas está disponível.

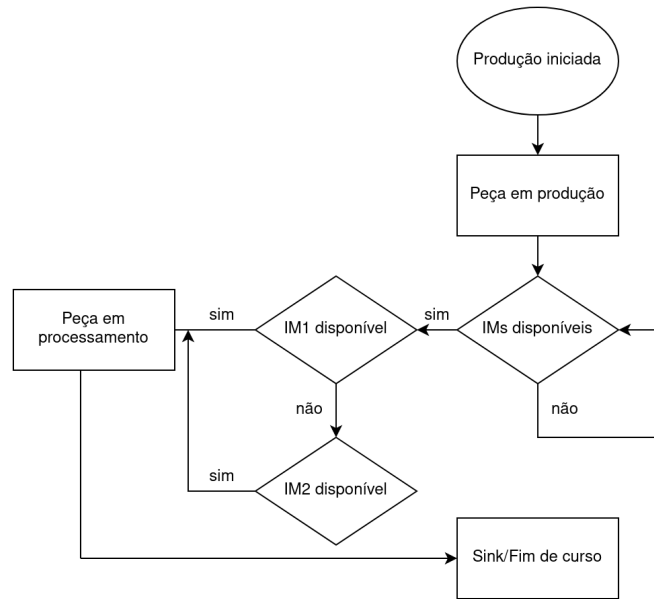


Figura 4.8: Circuito lógico que relaciona a operação de duas máquinas *indexing*.

Após esta atividade ser realizada, há a verificação de qual máquina *indexing* pode operar e a peça é transmitida. Após a finalização do procedimento, a peça é direcionada ao fim de curso.

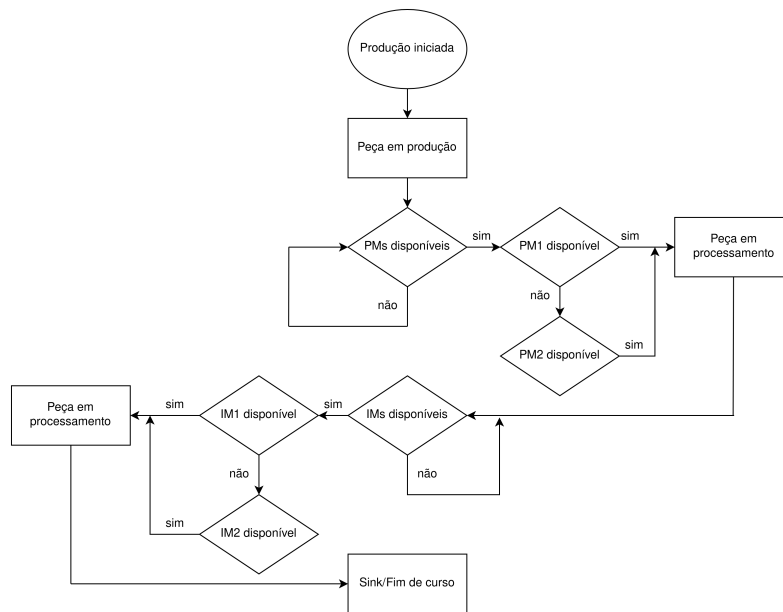


Figura 4.9: Circuito lógico: operação máxima de duas máquinas *indexing* e duas *punching*.

A partir do entendimento dessas lógicas de funcionamento do sistema de manufatura

foi desenvolvido o modelo de simulação no SIMIO. A visualização da configuração digital criada que representa o ambiente físico está disposta na Figura 4.10: os números 1 e 2 representam as máquinas *punching*, já os números 3 e 4 as máquinas *indexing*. Por fim, o número 5 representa a passadeira que leva ao final de curso. A partir desta estrutura foi possível aplicar os cenários envolvidos no *Digital Twin*.

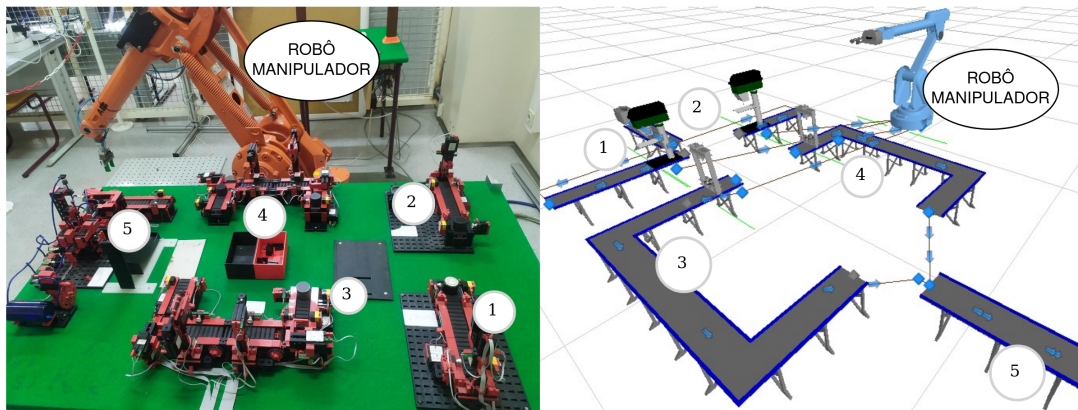


Figura 4.10: Representação 3D dos elementos de uma configuração produtiva.

Após o processamento das partes pelas devidas máquinas *punching*, o robô manipula as peças para os próximos nós da configuração, sendo estes localizados nas próximas máquinas, ou ainda nas passadeiras que levarão a uma possível finalização do processo produtivo, no estudo de caso realizado. Como em 4.11, ao realizar a aquisição da peça, o robô analisa a disponibilidade das máquinas *indexing* através do processo *2IndexUnavailableAlternative* para que as peças sejam processadas.

Na situação em que o usuário não tenha ativado ambas as máquinas, o robô redireciona as peças para a passadeira que leva ao fim de curso, sendo assim as peças atingem o nó do *Sink* e a estas encontram-se finalizadas. Caso alguma das máquinas esteja em funcionamento, o processo *robotProcess* faz com que o robô direcione as partes para as devidas máquinas, dependendo de qual esteja disponível naquele instante, pois os nós na entrada de cada máquina *indexing* ainda executam um processo denominado *Index1availableAlternative* para a primeira máquina e *Index2availableAlternative* para a segunda.

No caso de alguma destas máquinas ser consultada para a execução, apesar de não ter sido ativada, o nó é consultado e a peça é transferida para a máquina alternativa, de acordo com o passo *Transfer* implementado nos processos citados.

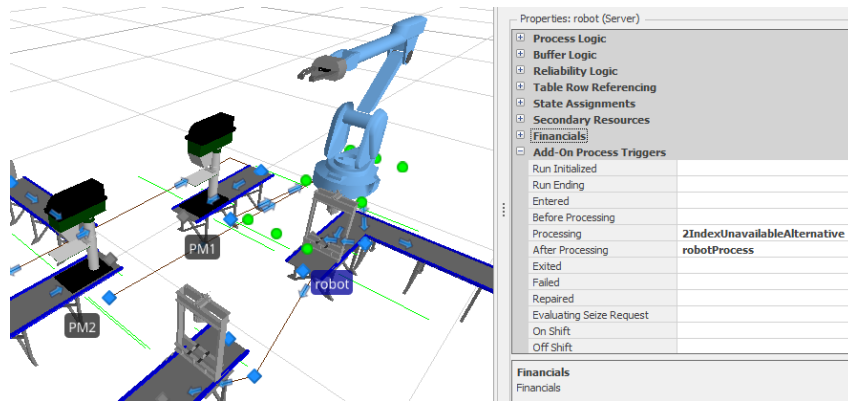


Figura 4.11: Processo de tomada de decisão do robô manipulador no ambiente de simulação.

O fluxograma da Figura 4.12 representa os processos relacionados ao funcionamento do robô.

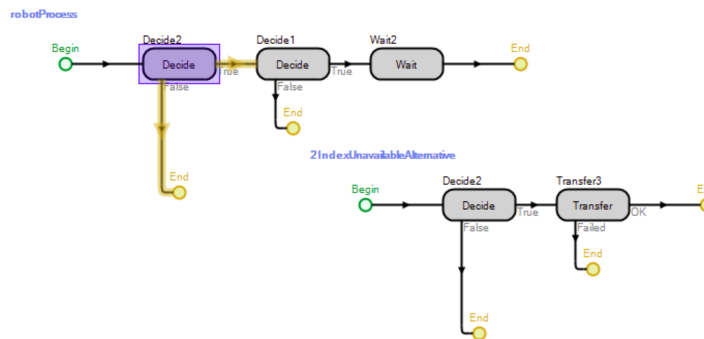


Figura 4.12: Processos relacionados quanto à tomada de decisão do robô manipulador.

Dois indicadores de grande relevância são *Cycle Time* e *Profit*. Portanto, para melhor entendimento destes fatores, as expressões a seguir detalham os parâmetros destes indicadores.

$$CycleTime = \frac{TimeReaching}{PartsProduced} \times 3600 \quad (4.1)$$

Como já citado, o software utiliza de processos de simulação discretos, e, portanto, escala temporal. Por conta disto, na expressão 4.1, foi possível atribuir às variáveis de estado o "tempo de entrega" das peças no fim de curso, ou seja, *TimeReaching*. Com o recurso da utilização da variável de estado já descrita *partsProduced*, e a taxa que converte os valores da escala temporal do software em valores tangíveis à realidade.

Para a obtenção do cálculo de lucro no estudo de caso realizado foi necessário considerar a receita adquirida a partir da quantidade de peças produzidas, o *Throughput* junto dos diversos custos de produção. Para que o software considerasse alguns custos externos, como valores para aquisição de matéria prima, uma taxa extra é aplicada. A expressão 4.2 mostra os parâmetros utilizados para os cálculos.

$$Profit = \frac{(partsProduced \times PartValue) - MachinesCost}{(partsProduced \times PartValue)} \times 100 \quad (4.2)$$

sendo que, para a variável que relaciona todos os custos de máquinas, se obtêm:

$$MachinesCost = robotCost + Index1Cost + PM1Cost + Index2Cost + PM2Cost \quad (4.3)$$

A expressão 4.3 reúne os dados de custo de cada máquina envolvida no processo, sendo estes valores baseados no que foi selecionado ainda no software SIMIO. Assim, a expressão 4.2 possui uma variável de custos representada pela equação 4.3, além da relação de receita. Desta forma, foi possível determinar a porcentagem de lucro envolvida nos processos específicos.

Após uma simulação, as variáveis predefinidas no SIMIO permitem que os dados sejam tratados pelos blocos lógicos de processamento, a fim de que um sistema *reports* seja finalizado a tempo de máquina. A Figura 4.13 mostra a secção *Pivot Grid*, a partir de onde dados que foram relacionados às variáveis de estado podem ser corretamente direcionados para que as lógicas fossem implementadas. Informações como *robot.ResourceState* são manipuladas pelos processos a fim de permitir que o sistema lógico utilizado na simulação atue devidamente, e ainda que os dados sejam ligados ao arquivo a ser exportado.

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
ModelEntity	PartA	[Population]	FlowTime	TimeInSystem	Minimum (Hou...	0,0136
					Observations	35,0000
			Throughput	NumberCreated	Total	36,0000
NumberDestroyed	Total	35,0000				
Path	Path1	[Travelers]	Content	NumberOnLink	Average	0,0498
					Maximum	1,0000
					FlowTime	TimeOnLink
			Maximum (Ho...	0,0012		
			Minimum (Hou...	0,0012		
			Throughput	NumberEntered	Total	18,0000
	NumberExited	Total		18,0000		
	Path1_1	[Travelers]		Content	NumberOnLink	Average
			Maximum			1,0000
			FlowTime			TimeOnLink
	Maximum (Ho...	0,0014				
	Minimum (Hou...	0,0014				
Throughput	NumberEntered	Total	18,0000			
	NumberExited	Total	18,0000			
	Path1_2	[Travelers]	Content	NumberOnLink	Average	0,0779
Maximum					1,0000	
FlowTime					TimeOnLink	Average (Hou...
Maximum (Ho...	0,0019					

Figura 4.13: Tabela relacional de dados de saída.

Após a finalização da simulação, com o auxílio dos processos criados, as informações são processadas e disponibilizadas em variáveis predefinidas e exportadas a aplicações externas, que utilizam destes dados para realizar outros tratamentos, geração de gráficos para monitorização, manipulação do escopo do nível de gestão de recursos, por exemplo, e exibição dos dados concretizados em *dashboards* para interação com o usuário final.

4.3 Desenvolvimento do Monitoramento

A disponibilização dos dados tratados em uma interface GUI aparece como a primeira interação entre máquina e o operador, que realiza assim o monitoramento do processo produtivo. Assim, com a possibilidade de melhorar a eficiência da manufatura, o usuário final utiliza os terminais do sistema supervisorio para selecionar as opções do nível produtivo que melhor atendem às suas necessidades.

A interface foi criada utilizando a linguagem de programação JavaScript na ferramenta de programação NodeRED, que permite a gestão de *flows* e blocos programáveis capazes de tratar e relacionar a informação, além de permitir a visualização por parte do usuário.

Com a capacidade de conectar *hardwares* e aplicações, a ferramenta foi utilizada como para completar a criação do DT. Por fim, esse utilizador realiza a tomada de decisão visualizando os dados apresentados na interface a partir da comparação entre os cenários simulados no *software* SIMIO e importandos para a ferramenta. A Figura 4.14 apresenta a forma como os blocos são disponibilizados no *flow* do NodeRED, com as linhas de programação em JavaScript.

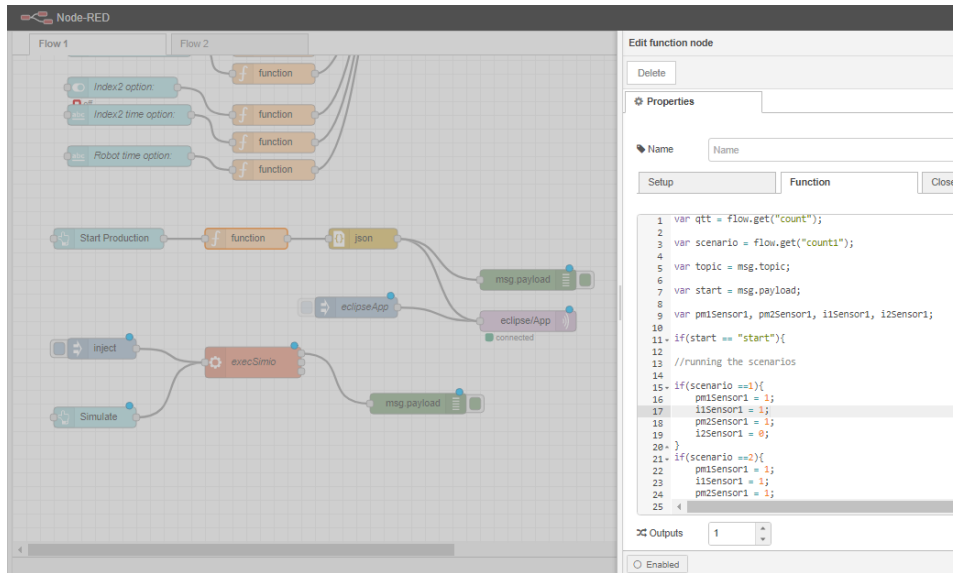


Figura 4.14: Representação dos blocos programáveis em NodeRed

Portanto, no caso de estudo, em um primeiro momento NodeRed recebe do software SIMIO as informações de uma primeira configuração do cenário produtivo, isto é, utilizando as informações de uma operação simples do sistema de manufatura, com uma máquina *punching* e uma máquina *indexing* durante um turno de produção.

São apresentadas para o cenário selecionado as características de tempo de operação, o número de peças produzidas, o tempo de ciclo para a produção, o lucro obtido, além da utilização de recursos. É apresentado ainda um aviso sobre o nível crítico de produção, uma vez que este fator tem influência direta quanto à tomada de decisão do operador. Os indicadores mostrados nessa *dashboard* são apresentados na Figura 4.15, e a partir deste ponto de situação o usuário pode optar por simular diferentes cenários para efeitos de comparação, buscando a melhor solução para as suas necessidades.

As informações apresentadas na parte superior correspondem ao valor selecionado para a peça, o estado das máquinas (ligado ou desligado), além do tempo de processamento de cada estação. Na parte inferior são apresentados os resultados da simulação no SIMIO para o cenário escolhido. São apresentados os dados de quantidade de peças produzidas no turno, o tempo de ciclo produtivo, o lucro obtido com a operação, além da utilização de recursos dos elementos de fabrico.

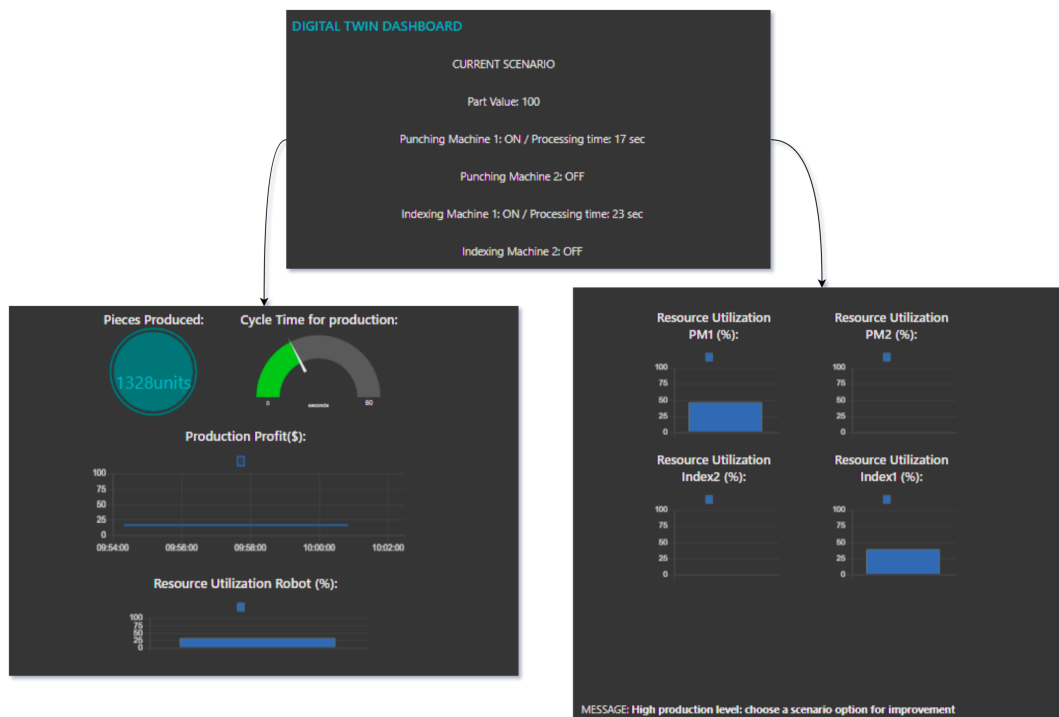


Figura 4.15: Dashboard com a configuração do cenário padrão

Após a atuação do utilizador ao realizar as simulações necessárias, este pode selecionar o botão "*Start Production*" que utiliza a comunicação MQTT entre NodeRED e a aplicação em JAVA para transmitir a informação resultante do cenário escolhido.

4.4 Sumário

Com a aquisição dos dados e disponibilização para a visualização no NodeRED, o utilizador pode selecionar na interface gráfica os novos cenários que serão simulados no *software*

SIMIO e comparados na *dashboard*. A partir disto, os dados resultantes da simulação são enviados para a aplicação JAVA, que possui conexão com o gêmeo físico, através da comunicação MQTT. Desta forma, a implementação do *Digital Twin* está completa e a tomada de decisão por parte do utilizador pode ser realizada.

Capítulo 5

Resultados e Discussões

Este capítulo apresenta os resultados obtidos a partir dos procedimentos descritos no Capítulo 4 para a implementação do DT. Considerações sobre as informações concluídas são elaboradas sobre a utilização do sistema de digitalização.

5.1 Aquisição e Comunicação de Dados

Como forma de integrar a informação do gêmeo físico baseado no sistema de manufatura foi utilizado o protocolo de comunicação Modbus/TCP com a criação da aplicação JAVA de aquisição na IDE Eclipse, permitindo que estes dados fossem exportados após a aquisição dos sinais por parte da IDE ao se conectar com o PLC. Por conta da flexibilidade presente no Eclipse e a possibilidade de utilização de *plug-ins* que permitem o seu direcionamento a variadas utilizações e interação com diversos *softwares*, o ambiente de desenvolvimento mostrou-se como uma escolha correta para a criação do DT.

Esta mesma aplicação permitiu que fosse desenvolvido o protocolo de comunicação MQTT de forma bidirecional, a fim de que informação com o ambiente de monitoramento fosse trocada. MQTT permitiu uma fácil configuração do sistema *broker* para que os recursos *publish* e *subscribe* fossem utilizados, fazendo com que a informação fosse transmitida de forma confiável para a aplicação JAVA. Este protocolo ainda permitiu compatibilidade com a ferramenta NodeRed, que pode receber e enviar dados através do

broker por conta dos blocos MQTT-IN e MQTT-OUT utilizados.

5.2 Cenários Simulados

A lógica de operação do *Digital Twin* descrita no Capítulo 4 permitiu que diversas opções de cenário fossem representadas, a fim de gerar formas de predição dos níveis produtivos. A Tabela 5.1 representa os cenários de simulação em que a configuração dos *layouts* de produção é modificado, se utilizando de adaptações e escalabilidade.

Após a seleção das características produtivas pelo utilizador, os resultados de simulação foram gerados. No caso das simulações realizadas, as estações operam com as mesmas características, isto é, o mesmo tempo de processamento das máquinas para cada peça produzida, junto do mesmo procedimento para a dinâmica do robô e das passadeiras em operação.

O primeiro cenário produtivo apresentado na interface de interação com o utilizador é denominado "Current Scenario" e é representado pela utilização de uma máquina *punching* e uma máquina *indexing*. Os seguintes cenários simulados apresentam as mesmas características para o valor do produto e tempos de operação, variando apenas a escalabilidade das máquinas. São relacionadas comparações entre os cenários, sendo estas dispostas na seguinte forma:

- **comparação 1:** primeiramente, na comparação entre **Cenário 1** e **Cenário 2** é apresentada a inserção de uma máquina *punching*;
- **comparação 2:** entre **Cenário 3** e **Cenário 4**, é escalada uma máquina *indexing*;
- **comparação 3:** entre **Cenário 5** e **Cenário 6**, há primeiramente a situação de produção mista com máquinas *punching* e *indexing*, em seguida é escalada uma máquina *indexing*;
- **comparação 4:** entre **Cenário 7** e **Cenário 8**, em que o operador opta por escalar mais uma máquina *indexing* quando já estava a atuar com duas *punching* e uma *indexing*, ou seja, operando com o funcionamento de quatro máquinas.

5.3 Simulação

A Tabela 5.1 apresenta as informações resultantes das simulações realizadas com base no *Digital Twin* desenvolvido. Na **comparação 1** é possível perceber que há diminuição no tempo de ciclo e por conta disto aumento no número de peças. Por conta dos custos operacionais da inserção de uma máquina, a elevação da produção ainda não é capaz de gerar um lucro muito superior, o que pode ser alterado com a relação de valor das peças.

No caso de estudo, o valor aplicado pelo operador na *dashboard* para a comercialização das partes, ou seja, a receita inferida para cada peça produzida, tem grande influência no resultado de lucratividade da operação porque há uma relação direta dos custos de produção com o lucro obtido para a situação simulada. Como exemplo, a Tabela 5.1 apresenta a simulação dos cenários para um valor fixo de receita por peça de 90 (euros, dólares, ou outra moeda, pois a simulação trata de relações matemáticas e porcentagens, independentemente da unidade monetária), e para a simulação com receita variável é apresentada a Tabela 5.2.

Desta forma, nesta situação em específico, o valor do lucro foi ligeiramente superior, o que leva a deduzir que o aumento no número de peças foi ínfimo em relação ao custo para inserção e operação de uma máquina, como é possível perceber no indicador que aponta a utilização de recursos. Neste sentido, para que o procedimento de escalabilidade de uma máquina *punching* seja lucrativo, esta deve atuar com melhor performance para que o *throughput* atinja uma escala mais útil. Além disto, caso o valor da peça seja de uma relevância maior, esta pequena melhoria no tempo de ciclo pode ser extremamente relevante.

Na **comparação 2**, como é percebido na Tabela 5.1, há uma diferença de maior relevância no tempo de ciclo por conta das características específicas destas máquinas no que diz respeito ao tempo de processamento, passando de 20.74 segundos com uma máquina, para 20.71 com duas. Desta forma, por conta da melhor eficiência deste processo, os outros indicadores da Tabela 5.1 mostram-se alinhados a esta situação, como o aumento da produtividade em 3 peças, além de melhor lucro apesar do aumento da utilização de

Indicadores	Throughput	Cycle Time	Resources Utilisation	Profit (%)
Cenário 1: 1 Punching Machine	1390	20.70	Robot: 69.4 PM1:37.4 PM2:0 IM1:0 IM2:0	12.5
Cenário 2: 2 Punching Machines	1391	20.69	Robot:69.5 PM1:15.4 PM2:27.6 IM1:0 IM2:0	12.51
Cenário 3: 1 Indexing Machine	1387	20.74	Robot:69.1 PM1:0 PM2:0 IM1:60 IM2:0	12.48
Cenário 4: 2 Indexing Machines	1390	20.71	Robot:69.4 PM1:0 PM2:0 IM1:23.7 IM2:37.4	12.49
Cenário 5: 1 Punching Machine & 1 Indexing Machine	1381	20.84	Robot:30.9 PM1:39.8 PM2:0 IM1:57.5 IM2:0	12.42
Cenário 6: 1 Punching Machine & 2 Indexing Machines	1384	20.79	Robot:33.2 PM1:41.6 PM2:0 IM1:33.2 IM2:20.2	12.45
Cenário 7: 2 Punching Machines & 1 Indexing Machine	1385	20.78	Robot:33.8 PM1:36.9 PM2:39.2 IM1:0 IM2:58.7	12.45
Cenário 8: 2 Punching Machines & 2 Indexing Machines	1384	20.80	Robot:37 PM1:39.8 PM2:38.6 IM1:33.6 IM2:19.4	12.44

Tabela 5.1: Resultados do estudo de caso: relação com utilização de recursos.

recursos.

Similarmente, na **comparação 3**, como pode-se notar na Tabela 5.1, o nível de peças produzidas passa de 1381 para 1384. Assim, ocorre aumento no lucro, por conta da diminuição relevante ocorrida no tempo de ciclo, capaz de superar o aumento de custos gerados pela escalabilidade de uma máquina, notados em *Resource Utilisation*.

Já na situação da **comparação 4**, como também é visto na 5.1, há a situação adversa de diminuição de 1385 para 1384 peças e, conseqüentemente, o lucro obtido tem reflexo na relevância que os custos de operação empregam. Neste sentido, para o valor de receita selecionado, é percebida a queda na lucratividade. Nesta situação é possível notar que não há a necessidade de escalar outra máquina *indexing* por conta dos diferentes tempos de operação entre as máquinas, e apenas uma máquina seria capaz de processar as peças eficientemente.

Como já mencionado, os valores obtidos de receita para cada cenário de produção tem influência direta quanto à tomada de decisão se aquela situação é favorável ou não. A Tabela 5.1 utilizou em valor fixo de receita, enquanto a Tabela 5.2 mostra resultados para diferentes valores de receita sobre os cenários já aplicados, neste caso (*Product Value*). Os indicadores apresentados para efeitos de comparação são a quantidade de partes produzidas e o lucro obtido, que nesta situação sofre alterações de acordo com o valor arrecadado de receita.

Como pode-se perceber na Tabela 5.2 nas representações do **Cenário 1**, a situação para utilização de uma máquina *punching* sofre a queda drástica de lucratividade, sendo esta de 5.56%, apenas com a redução do valor da peça de 100 para 60. Situação semelhante ocorre para os outros cenários apresentados, o que demonstra a importância da relação de negócios também no ambiente produtivo. O **Cenário 5** sofre queda de 5.52%, com queda de valor das peças de 80 para 40. O lucro no **Cenário 7** cai 5.54% com os valores passando de 70 para 30, enquanto o **Cenário 8** cai 4.18% por conta da desvalorização de 50 para 20.

Assim, a partir dos resultados obtidos nas simulações realizadas, o planejamento de disponibilização de recursos pode ser realizado e a tomada de decisão sobre os ambientes

Indicadores	Throughput	Product Value	Profit (%)
Cenário 1: 1 Punching Machine	1390	100	13.89
Cenário 1: 1 Punching Machine	1390	60	8.33
Cenário 5: 1 Punching Machine & 1 Indexing Machine	1381	80	11.04
Cenário 5: 1 Punching Machine & 1 Indexing Machine	1381	40	5.52
Cenário 7: 2 Punching Machines & 1 Indexing Machine	1381	70	9.65
Cenário 7: 2 Punching Machines & 1 Indexing Machine	1381	30	4.11
Cenário 8: 2 Punching Machines & 2 Indexing Machines	1384	50	6.90
Cenário 8: 2 Punching Machines & 2 Indexing Machines	1384	20	2.72

Tabela 5.2: Resultados do caso de estudo: relação com a receita.

de manufatura pode ser contemplada de forma mais ampla e eficaz.

5.4 Monitoramento

O utilizador realiza interação apenas com a HMI desenvolvida em NodeRED. Desta forma, este operador não tem acesso ao ambiente de simulação, estando restrito aos dados apresentados na interface. Assim, este realiza a seleção de algumas opções de cenários e parte para a tomada de decisão a partir das informações obtidas.

A Figura 5.1 mostra a situação simulada com base em um dos cenários da Tabela 5.1, e representa a escolha do usuário por duas máquinas *punching* a operar. Nesta situação, é possível perceber um lucro obtido de 13% em comparação a situação anterior de apenas uma máquina que contava com 16%. A partir do escopo produtivo selecionado, o utilizador pode optar por aquele que lhe trouxer melhor lucro, menos custos, menor tempo de ciclo ou maior flexibilidade.

Na situação apresentada na Figura 5.2, novas características foram selecionadas pelo utilizador, como menor *Part Value* quando comparada ao cenário anterior e maior número de máquinas em operação. Por conta disto há um maior *Cycle Time*, de 20.85 segundos, enquanto o cenário anterior contava com 20.70 segundos. Sendo assim, há um menor número de peças produzidas, além de um cenário com maior escalabilidade, ou seja, maior número de máquinas em operação com tempos produtivos diferentes. Nesta situação, foi concluído um lucro de 5% sobre a produção.

O caso simulado mostra que os novos valores de indicadores são dispostos na interface, e a partir daí o operador realiza a tomada de decisão, que pode incluir:

- Aumentar o valor da peça por conta da sua maior complexidade ao ser manipulada por uma máquina *indexing*.
- Utilizar as mesmas características da PM2 na PM1, ou seja, atualizar a eficiência temporal da máquina 1 para melhoria do processo, reduzindo o tempo de ciclo.

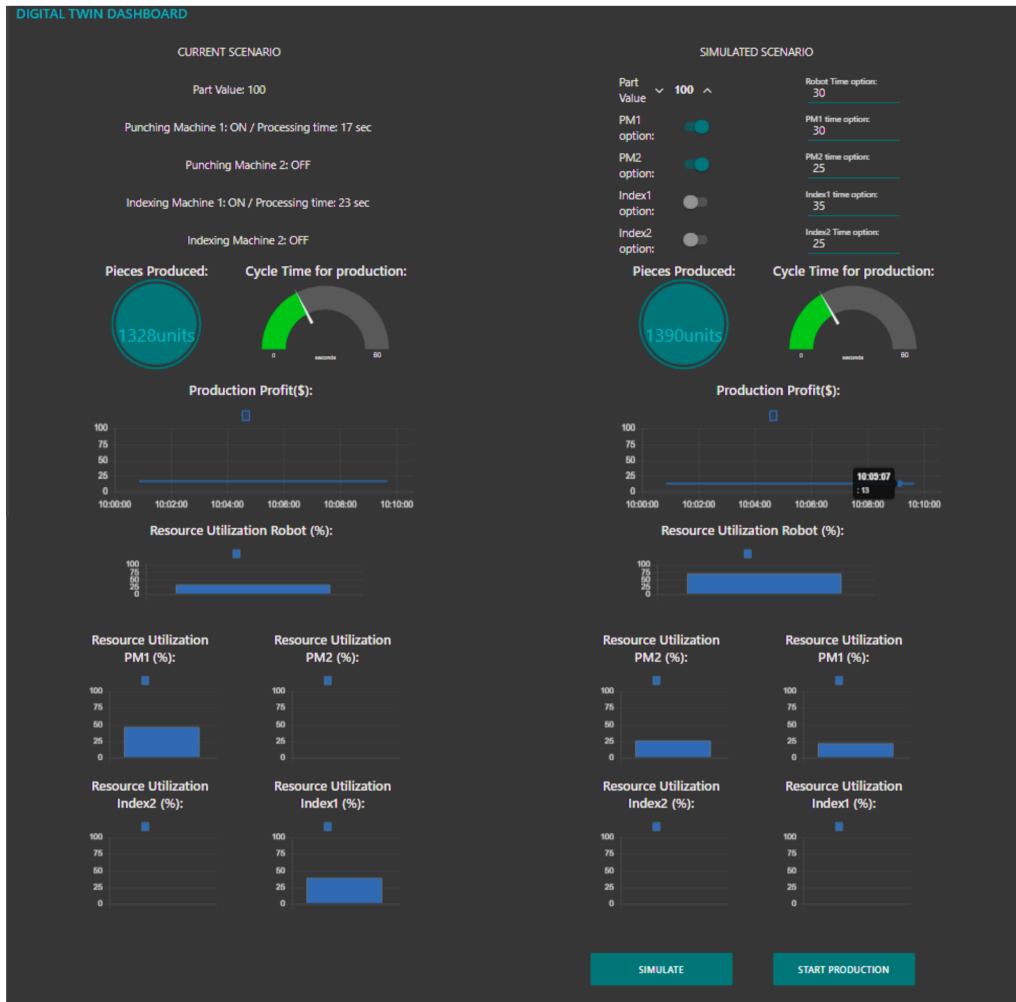


Figura 5.1: Dashboard: comparação entre a configuração básica e duas máquinas *punching*

- Simular um diferente cenário, dispondo de apenas uma máquina *punching*, para comparar o impacto de utilização dos recursos no que diz respeito aos custos produtivos.

5.5 Dificuldades de Desenvolvimento

Por conta da disponibilização de recursos financeiros, os softwares utilizados contaram com licenças limitadas ou abertas. Em geral, estes ambientes de desenvolvimento contam com menor número de ferramentas, e por conta disto diversas adaptações tiveram que ser

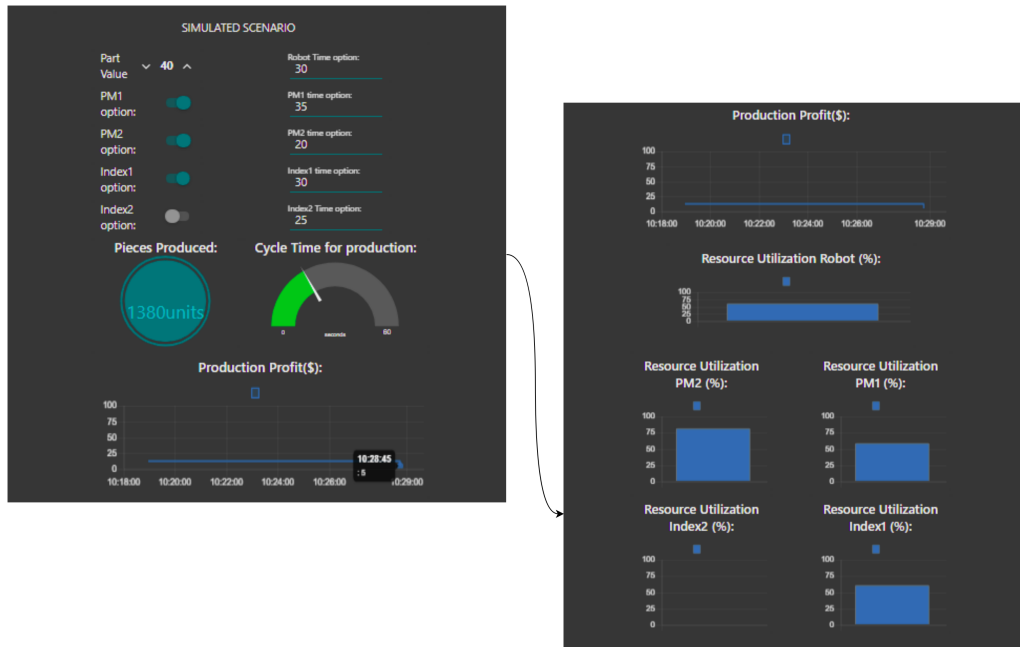


Figura 5.2: Dashboard: configuração para duas máquinas *punching* e uma máquina *indexing*

realizadas para que o funcionamento do *Digital Twin* ocorresse de acordo. Como no caso do SIMIO, diversas funções que poderiam ter sido implementadas tiveram que ser adaptadas, levando a um trabalho arduo e que demandou imenso tempo de desenvolvimento. Por não estar adaptado à utilização de métodos de comunicação com outras aplicações, em uma situação de *remote API*, a forma de transmissão de informação entre o SIMIO e outras aplicações foi prejudicada. Desta forma, a utilização de uma licença completa para a implementação do sistema de simulação do *Digital Twin* ou ainda a utilização de outro *software* simulador é de grande valia, sendo que outras opções foram apresentadas nesta dissertação.

Em relação ao ambiente físico, as células Fischertechnik não representam uma linha de montagem propriamente dita, embora mostrem-se como uma opção valiosa para a realização de simulações. Neste sentido, adaptações entre um cenário real e a situação do conjunto de estações precisaram ser realizadas. Há ainda a rusticidade do robô ABB IRB 1400, que apenas permite a sua programação *offline*, ou seja, é necessária a parada do sistema produtivo para adaptações no código do robô.

5.6 Sumário

Após a comunicação dos dados de forma eficiente e disponibilização para a visualização no NodeRED, o utilizador selecionou na interface gráfica os cenários necessários para o seu sistema de manufatura. Com a capacidade de flexibilidade das características do sistema, diversos resultados foram obtidos após simulação no *software* SIMIO, e estes dados apresentados na *dashboard*. A partir disto, a informação concluída foi enviada à aplicação JAVA, que está ligada ao gêmeo físico, através de comunicação MQTT.

Capítulo 6

Conclusões e Trabalhos Futuros

Este capítulo possui a finalidade de reunir as principais percepções sobre o estudo realizado, colocando de forma resumida as considerações gerais.

Conclusões

Os ambientes flexíveis de manufatura passam por uma transformação digital no contexto da Indústria 4.0. A aplicação do *Digital Twin* a estes ambientes apresenta-se como uma solução para questões de análises do sistema produtivo e tomada de decisão. Por conta do grande interesse atraído pelo DT, o número de informação gerada sobre o tema é crescente.

O caso de estudo foi concluído com o alcance do objetivo geral a princípio proposto, sendo este de realizar a otimização de um sistema flexível de fabrico no âmbito da indústria 4.0 através da aplicação de um DT. O laboratório LCAR possibilitou que sua estrutura do sistema produtivo fosse utilizada para que fosse completada a criação do DT, contando com um robô ABB e estações compostas por células Fischertechnik. A partir dos sinais adquiridos e integrados no PLC, foi possível comunicar o gêmeo físico com o modelo digital.

O *software* simulador SIMIO permitiu que cenários fossem desenvolvidos para que as possibilidades do sistema de manufatura fossem representadas. A partir das simulações

realizadas de acordo com os objetivos do utilizador, as informações foram representadas em uma interface gráfica para visualização e tomada de decisão. Este ambiente permite ao utilizador interagir com a simulação e comunicar as informações, com possibilidade de escalabilidade dos processos, ou seja, da alteração e até adição de estações de processamento.

De forma geral este trabalho contribui com a Indústria 4.0 ao possibilitar melhorias no setor de manufatura a partir da integração de um sistema físico com um sistema digital. Com o *Digital Twin* desenvolvido, o operador do sistema flexível é capaz realizar a tomada de decisão por conta dos níveis de predição estabelecidos.

No que diz respeito ao âmbito científico, foi desenvolvido um caso de estudo específico sobre o trabalho e a partir daí foi escrito um artigo que tem relevância para a Indústria 4.0. Este artigo foi aceito pela conferência SOHOMA 2020 (*10th SOHOMA European Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*).

Trabalhos Futuros

O estudo realizado levou em conta as tecnologias dispostas em laboratório, sendo que neste caso, embora em pleno funcionamento, alguns módulos físicos encontram-se já desatualizados, como no caso do robô ABB. O trabalho foi desenvolvido com a consideração de que os módulos pudessem ser substituídos por outros mais eficientes, como na *dashboard* no NodeRED que permite a alteração do tempo de processamento das máquinas para atualizar a eficiência do conjunto. Além disto, ao se utilizar tecnologias mais modernas, operações em tempo real podem ser concluídas sem a necessidade de parada da produção para programação do robô com os disquetes. Neste caso, ainda seria possível criar o modelo digital do robô para que este seja incluído de forma eficiente no DT, monitorizando os seus detalhes de operação, como eixos e garras.

Por conta da disponibilização de recursos financeiros, os softwares utilizados contaram com licenças limitadas, como no caso do SIMIO. Para que os trabalhos desenvolvidos acompanhem outras contribuições no setor de IoT, é de grande valia a utilização de

softwares mais completos, adaptados às opções de comunicação de dados disponíveis no mercado e com menor latência na comunicação. SIMIO não conta com a possibilidade de comunicação remota com outras aplicações, o que seria de grande importância para obter simulações em tempo real para o DT.

Foi lançado mão do desenvolvimento de alguns indicadores sobre o DT, que no caso de uma operação em tempo real poderia ser expandido a diversas outras características de um sistema produtivo como tempo médio entre falhas, tempo médio entre reparos, número de paradas na produção, além de uma abordagem mais aprofundada quanto à eficiência e utilização dos recursos.

Por fim, a partir da interface desenvolvida em NodeRed, o *Digital Twin* contou com a tomada de decisão por parte do utilizador, o que em trabalhos futuros pode ser realizado com a tomada de decisão automática ao inserir níveis de inteligência artificial através de algoritmos no sistema ciberfísico.

Bibliografia

- [1] L. L. Li Da Xu Eric L. Xu, “Industry 4.0: state of the art and future trends,” *International Journal of Production Research*, 2017.
- [2] J. C. R. Diogo Pimenta e J. F. Oliveira, “The Social Impact of the Use of Cyber-Physical Systems in Manufacturing: An Initial Approach,” *Proc. Intel European Research & Innovation Conf.*, pp. 196, Leipzig, out. de 2011.
- [3] K. Schwab, “The Fourth Industrial Revolution,” *Edipro*, 2017.
- [4] M. Negri Fumagalli, “A Review of the Roles of Digital Twin in CPS-based Production Systems,” *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017, 27th International Conference on Flexible Automation and Intelligent Manufacturing, jun. de 2017.
- [5] W. T. Jones A.T. Romero D., “Modeling agents as joint cognitive systems in smart manufacturing systems,” *Proc. Intel European Research*, out. de 2018.
- [6] J. M. D. S. H. K. Min Xu, “The Fourth Industrial Revolution: Opportunities and Challenges,” *International Journal of Financial Research*, 2018.
- [7] E. E. J. A. Manuel Sanchez, “Industry 4.0: survey from a system integration perspective,” *International Journal of Production Research*, 2020.
- [8] L. Parrot, “Industry 4.0 and the Digital Twin,” *Delloite University Press*, mai. de 2017.
- [9] G. Doyle, “Modeling, Simulation, Information, Technology Processing,” abr. de 2010.

- [10] M. Z. Fei Tao Qinglin Qi, “Digital twin-driven product design framework,” *International Journal of Production Research*, 2018.
- [11] K. Henning, “Recommendations for implementing the strategic initiative Industrie 4.0,” *National Academy of Science and Engineering*, 2013.
- [12] G. Stargel, “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles,” *Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin*, abr. de 2012.
- [13] P. L. Flavia Pires Ana Cachada, “Digital Twin in Industry 4.0: Technologies, Applications and Challenges,” *IEEE Conference on Industrial Informatics*, jul. de 2019.
- [14] Rohan, “Digital Twin Market Worth 15.66 Billion USD by 2023,” *MarketsandMarkets*, [Online]. Available: <https://www.prnewswire.com/in/news-releases/digital-twin-market-worth-1566-billion-usd-by-2023-642374603.html>. [Accessed: 30-July-2020]., 2017.
- [15] S. Singh, “Digital Twin Market worth \$35.8 billion by 2025,” *MarketsandMarkets*, [Online]. Available: <https://www.marketsandmarkets.com/PressReleases/digital-twin.asp>. [Accessed: 30-July-2020]., 2019.
- [16] P. Leitao, “A holonic disturbance management architecture for flexible manufacturing systems,” *International Journal of Production Research*, 2010.
- [17] H.-A. K. Jay Lee Behrad Bagheri, “Cyber-Physical Systems architecture for Industry4.0-based manufacturing systems,” *Proc. Intel European Research & Innovation Conf.*, pp. 196, Leixlip, dez. de 2014.
- [18] H. P. Paridhika Kayal, “A comparison of IoT application layer protocols through a smart parking implementation,” *Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017.
- [19] K. R. James Kurose, “Computer Networking: A Top-Down Approach,” *Pearson*, 2000.
- [20] T. Rando, “A State of the Art Report of CAD/CAM/CIM Systems Technologies for the U.S. Shipbuilding Industry,” *Naval Surface Warfare Center*, 2002.

- [21] Cedri-Website, “http://cedri.ipb.pt/aboutus_laboratories.html,” Acessado em: 18-07-2020.
- [22] Modbus-Website, “<https://www.modbus.org/>,” Acessado em: 15-07-2020.
- [23] Eclipse-Website, “<https://www.eclipse.org/>,” Acessado em: 16-07-2020.
- [24] Al-Fuqaha, “The IoT Elements Source,” *IEEE COMMUNICATION SURVEYS VOL. 17, NO. 4, FOURTH QUARTER 2015*, 2015.
- [25] R. T. Fieldin, “Architectural Styles and the Design of Network-based Software Architectures,” *Dissertation*, 2000.
- [26] A. Dürkop, “Embedding Openness in the Design Process of Web-based Learning Arrangements,” *OPINE 2019, Frankfurt*, 2019.
- [27] B. P. Meena Singh, “Secure MQTT for Internet of Things,” *Fifth International Conference on Communication Systems and Network Technologies*, 2015.
- [28] Z. Roeder Frazier, “INTRODUCTION TO SIMIO,” *Proceedings of the 2016 Winter Simulation Conference*, 2016.
- [29] K. K. Nadège Benkamouna Waguih ElMaraghyb Anne-Lise Huyeta, “Architecture Framework for Manufacturing System Design,” *Procedia CIRP*, 2014.
- [30] J. V. P. de Almeida, “Virtualização de um Sistema Flexível de Fabrico no âmbito da Indústria 4.0,” *Dissertação de Mestrado*, 2020.

**Proposta de Dissertação / Projeto
Mestrado em Engenharia Industrial
2019/2020**

Título Provisório: Otimização de sistema flexível de fabrico no âmbito da Indústria 4.0

Aluno de mestrado: Mathaeus Lukas Franklin R T M de Souza

Orientador: Paulo Leitão

Coorientador (UTFPR): Joaquim Mira

Principais objetivos a atingir:

Pretende-se realizar a implementação de técnicas de otimização do processo produtivo de um sistema flexível de fabrico, utilizando os dados coletados dos sensores/atuadores do sistema em tempo real em ambientes físico e virtual e algoritmos em linguagens de programação que permitam a coleta, análise e visualização dos dados coletados de forma a otimizar o sistema produtivo. Neste ambiente será possível realizar a interação da planta estudada com outros mecanismos também disponíveis, permitindo assim a otimização através de novos indicadores e comparação do novo sistema elaborado com os sistemas antigos.

Resultados esperados:

Pretende-se com esta proposta realizar o desenvolvimento e implementação de técnicas de otimização do processo produtivo utilizando a virtualização de um sistema flexível de fabrico existente no laboratório de controlo, automação e robótica.

Este trabalho está inserido nas atividades investigação, desenvolvimento e inovação que se encontram a ser desenvolvidas no CeDRI – Centro de Investigação e Robótica Inteligente (<http://www.cedri.ipb.pt>).

Calendarização das fases do trabalho:

O desenvolvimento da presente proposta de trabalho será realizado através da execução das seguintes etapas:

1. Familiarização com o conceito de *digital twin* e *machine learning*, permitindo assim o conhecimento das ferramentas utilizáveis e desenvolvimento de habilidades de aplicação sobre o sistema proposto (M1)
2. Estudo do caso de estudo, nomeadamente do sistema flexível de fabrico, do Digital Twin e de possíveis interações que possam ser realizadas com a máquina UR3, permitindo assim comparações com outros sistemas robóticos já disponíveis e verificando modelagens capazes de serem aplicadas em variados sistemas com interação da máquina (M2)
3. Elaboração do sistema de análise de dados e lógica de otimização através de novos indicadores e novas interações realizadas junto da planta (M3 – M4)
4. Implementação do algoritmo de otimização a partir de indicadores, como tempo medio entre falhas, setup, produtividade, entre outros. Comparação do sistema atual com o antigo, detectando possíveis falhas, elaborando melhorias, expandindo para maiores possibilidades de interação com o ambiente na planta (M5-M6)
5. Testes e validação do protótipo (M7 – M9)
6. Escrita da dissertação e defesa final do trabalho (M10 - M11)

Palavras-chave: Digital Twin, IoT, otimização, machine learning, Indústria 4.0.

Infraestruturas a utilizar:

Está prevista a utilização do sistema flexível de fabrico do CeDRI.